F 10

```
IIIIIIIII   NNN         NNN   SSSSSSSSSSSS  TTTTTTTTTTTTTTT     AAAAAAAAA   LLL
IIIIIIIII   NNN         NNN   SSSSSSSSSSSS  TTTTTTTTTTTTTTT     AAAAAAAAA   LLL
IIIIIIIII   NNN         NNN   SSSSSSSSSSSS  TTTTTTTTTTTTTTT     AAAAAAAAA   LLL
   III      NNN         NNN SSS                  TTT       AAA         AAA LLL
   III      NNN         NNN SSS                  TTT       AAA         AAA LLL
   III      NNN         NNN SSS                  TTT       AAA         AAA LLL
   III      NNNNNN      NNN SSS                  TTT       AAA         AAA LLL
   III      NNNNNN      NNN SSS                  TTT       AAA         AAA LLL
   III      NNNNNN      NNN SSS                  TTT       AAA         AAA LLL
   III      NNN   NNN   NNN   SSSSSSSSS          TTT       AAA         AAA LLL
   III      NNN   NNN   NNN   SSSSSSSSS          TTT       AAA         AAA LLL
   III      NNN   NNN   NNN   SSSSSSSSS          TTT       AAA         AAA LLL
   III      NNN      NNNNNN            SSS       TTT       AAAAAAAAAAAAAAA LLL
   III      NNN      NNNNNN            SSS       TTT       AAAAAAAAAAAAAAA LLL
   III      NNN      NNNNNN            SSS       TTT       AAAAAAAAAAAAAAA LLL
   III      NNN         NNN            SSS       TTT       AAA         AAA LLL
   III      NNN         NNN            SSS       TTT       AAA         AAA LLL
   III      NNN         NNN            SSS       TTT       AAA         AAA LLL
IIIIIIIII   NNN         NNN SSSSSSSSSSSS         TTT       AAA         AAA LLLLLLLLLLLLLLL
IIIIIIIII   NNN         NNN SSSSSSSSSSSS         TTT       AAA         AAA LLLLLLLLLLLLLLL
IIIIIIIII   NNN         NNN SSSSSSSSSSSS         TTT       AAA         AAA LLLLLLLLLLLLLLL
```

E 13

```
**FILE**ID**INSCREATE
```

```
  IIIIII    NN      NN    SSSSSSSS    CCCCCCCC  RRRRRRRR    EEEEEEEEEE    AAAAAA    TTTTTTTTTT  EEEEEEEEEE
  IIIIII    NN      NN    SSSSSSSS    CCCCCCCC  RRRRRRRR    EEEEEEEEEE    AAAAAA    TTTTTTTTTT  EEEEEEEEEE
    II      NN      NN  SS          CC          RR      RR  EE          AA      AA      TT      EE
    II      NN      NN  SS          CC          RR      RR  EE          AA      AA      TT      EE
    II      NNNN    NN  SS          CC          RR      RR  EE          AA      AA      TT      EE
    II      NNNN    NN  SS          CC          RR      RR  EE          AA      AA      TT      EE
    II      NN  NN  NN    SSSSSS    CC          RRRRRRRR    EEEEEEEE    AA      AA      TT      EEEEEEEE
    II      NN  NN  NN    SSSSSS    CC          RRRRRRRR    EEEEEEEE    AA      AA      TT      EEEEEEEE
    II      NN    NNNN          SS  CC          RR  RR      EE          AAAAAAAAAA      TT      EE
    II      NN    NNNN          SS  CC          RR  RR      EE          AAAAAAAAAA      TT      EE
    II      NN      NN          SS  CC          RR    RR    EE          AA      AA      TT      EE          ....
    II      NN      NN          SS  CC          RR    RR    EE          AA      AA      TT      EE          ....
  IIIIII    NN      NN  SSSSSSSS      CCCCCCCC  RR      RR  EEEEEEEEEE  AA      AA      TT      EEEEEEEEEE  ....
  IIIIII    NN      NN  SSSSSSSS      CCCCCCCC  RR      RR  EEEEEEEEEE  AA      AA      TT      EEEEEEEEEE  ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE INSCREATE (                           ! Create KFE entry
0                              IDENT = 'V04-000',
0                              ADDRESSING_MODE(EXTERNAL = GENERAL)
0                      ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 ! FACILITY:  Install
1 !
1 ! ABSTRACT:
1 !
1 !       This module executes the CREATE, REPLACE and DELETE options on INSTALL
1 !
1 ! ENVIRONMENT:
1 !
1 !       VAX/VMS operating system.
1 !
1 ! AUTHOR:  Bob Grosso, April 1983
1 !
1 ! Modified by:
1 !
1 !
1 !       V03-023 MSH0065         Michael S. Harvey       16-Jul-1984
1 !               Don't allow privileged or execute only images to have
1 !               transfer arrays pointing to SYS$IMGSTA.
1 !
1 !       V03-022 MSH0061         Michael S. Harvey       5-Jul-1984
1 !               Add EXEONLY support.
1 !
1 !       V03-021 MSH0057         Michael S. Harvey       26-Jun-1984
1 !               Store WRITEABLE attribute in KFE so that it can be
1 !               propagated across a REPLACE command along with all
1 !               the other attributes.
```

```
1 !
1 !     V03-020 MSH0047         Michael S. Harvey       11-May-1984
1 !             Add some image header validation checks for images being
1 !             installed with resident headers since such checks will
1 !             not be done in the image activator for these cases.
1 !
1 !     V03-019 MSH0046         Michael S. Harvey       11-May-1984
1 !             Calculate an effective IDENT for shareable compatibility
1 !             mode global sections, that is, an IDENT that can be
1 !             used by the AME. Also, don't attempt to determine the
1 !             state of being "shareable" for C-mode images by applying
1 !             the native mode test for that state.
1 !
1 !     V03-018 MSH0038         Michael S. Harvey       30-Apr-1984
1 !             Correct parameter definition in call to IMG$DECODE_IHD
1 !             so that compatibility mode images are correctly recognised.
1 !             Also, update ALIAS check to conform to the image activator's
1 !             check. Also, correctly set SHM when attempting to install
1 !             images with shared memory global sections.
1 !
1 !     V03-017 MSH0033         Michael S. Harvey       16-Apr-1984
1 !             Back out part of MSH0030 below. Turns out that we only
1 !             want to change the page write access mode, while leaving
1 !             the page ownership as USER instead of EXEC.
1 !
1 !     V03-016 MSH0028         Michael S. Harvey       11-Apr-1984
1 !             Maximum shared count now has meaning even for non-shareable
1 !             images. Initialize the count in a more general way.
1 !
1 !     V03-015 MSH0030         Michael S. Harvey       9-Apr-1984
1 !             Set up page ownership for protected images correctly.
1 !
1 !     V03-014 MSH0028         Michael S. Harvey       9-Apr-1984
1 !             Correctly set initial maximum shared count for shareable
1 !             known file images.
1 !
1 !     V03-013 MSH0024         Michael S. Harvey       31-Mar-1984
1 !             Don't attempt to create global sections for compatibility
1 !             mode tasks which are not built shareable (TKB /MU).
1 !             Also, don't set SHARED or HDRRES bits if they shouldn't
1 !             be set. This prevents later screwups in case the known
1 !             file image is deleted. Also, clean up warning to c-mode
1 !             users that resident headers are not allowed for such images.
1 !
1 !     V03-012 MSH0022         Michael S. Harvey       15-Mar-1984
1 !             Eliminate middle brackets from root directory spec.
1 !             Also, correct logic which flags the shared memory state.
1 !             Also, clarify NOGBLSEC message so it's more useful.
1 !
1 !     V03-011 MSH0018         Michael S. Harvey       7-Mar-1984
1 !             Remove obsolete check for maximum file name length. It's
1 !             obsolete now that global sections support 39 character
1 !             file names.
1 !
1 !     V03-010 MSH0017         Michael S. Harvey       7-Mar-1984
1 !             Prevent pool loss when trying to install an image for
1 !             which another version of the image is already installed.
```

```
1 !
1 !             V03-009 MSH0015         Michael S. Harvey       6-Mar-1984
1 !                     Warn user when installing a shareable image and no global
1 !                     sections can be created.
1 !
1 !             V03-008 MSH0004         Michael S. Harvey       13-Feb-1984
1 !                     Don't reject long image names. Also, add support of long
1 !                     global section names.
1 !
1 !             V03-007 MSH0003         Michael S. Harvey       27-Jan-1984
1 !                     Prevent crash caused by eventual system service execution
1 !                     while IPL is incorrectly left at ASTDEL.
1 !
1 !             V03-006 BLS0256         Benn Schreiber          3-Jan-1984
1 !                     Correct calls to allocate paged pool to check for errors
1 !                     so that system doesn't crash.  Convert square brackets
1 !                     to angle brackets in KFD list.  Don't allocate new KFD
1 !                     until we are ready to enter the KFE.
1 !
1 !             V03-005 RPG0005         Bob Grosso              01-Aug-1983
1 !                     Change Global section ident to be something other
1 !                     than zero for non shareable images.
1 !                     Set IPL to ASTDEL to ensure process is not deleted
1 !                     with pool allocated but not yet connected to list.
1 !                     Also comment code.
1 !
1 !             V03-004 RPG0004         Bob Grosso              July 25, 1983
1 !                     Count entries to assist listing.
1 !
1 !             V03-003 RPG0003         Bob Grosso              July 20, 1983
1 !                     Correct call to MMG$RET_BYT_QUOTA.
1 !
1 !             V03-002 RPG0002         Bob Grosso              July 19, 1983
1 !                     Create protected global sections in user mode instead
1 !                     of exec mode.
1 !                     Set the SHRWCB bit in the WCB and call MMG$RET_BYT_QUOTA.
1 !                     To return byte quota since file is being opened for everyone.
1 !
1 !             V03-001 RPG0001         Bob Grosso              July 7,1983
1 !                     Reduce items on kernel stack
1 !--
1
1 !
1 ! Include files
1 !
1
1 LIBRARY 'SYS$LIBRARY:LIB';                            ! VAX/VMS system definitions
1
1 REQUIRE 'SRC$:INSPREFIX.REQ';
1 REQUIRE 'SHRLIB$:IMGMSGDEF.R32';                      ! Message codes for the image header decode routines
1 REQUIRE 'LIB$:INSDEF.R32';                           ! Contains definition of INSTALL flags longword
1 REQUIRE 'LIB$:RSXLBLDF.R32';                         ! Contains field offsets for compatability mode image header
```

```
1 %SBTTL  'Declarations';
1
1 LINKAGE
1     JSB_0 = JSB (REGISTER = 0),                  ! for MMG$RET_BYTE_QUOTA
1
1     JSB_0_G1 = JSB (REGISTER = 0) :              ! for IOC$VERIFYCHAN
1         GLOBAL (ccb = 1) NOPRESERVE (2,3),
1
1     JSB_G1_G2 = JSB :                            ! Allocate pool
1         GLOBAL (length = 1, entry_block = 2)
1         NOPRESERVE (3),
1
1     JSB_G1_G2_3 = JSB (REGISTER = 3) :           ! Allocate memory in P1 space
1         GLOBAL (length = 1, entry_block = 2),
1
1     JSB_9_G10_G11 = JSB (REGISTER = 9) :         ! MMG$GSDTRNLOG
1         GLOBAL (SHRMEMNAM = 10, GSDNAM = 11);
1
1 !
1 ! Table of contents
1 !
1 FORWARD ROUTINE
1     INS_CREATE,                                  !
1     CREATE,                                      !
1     ALLOC_PAGED,                                 ! Allocate from paged pool
1     FIND_KFD,                                    !
1     BUILD_KFD : NOVALUE,                         ! Build a Known file Device, directory block
1     ENTER_KFE,                                   ! Insert the Known File Entry into the Hash list and KFD lis
1     VERIFY_CHANNEL,                              !
1     CHECK_SHMIDENT,                              ! Check if global sections should be in shared memory
1     INS$BLD_GBLSECNAM;                           ! Build the global section name with the _nnn suffix
1
1 EXTERNAL ROUTINE
1     INS$EXECUTE_IN_KRNL_WITH_W_LOCK,
1     INS$CNVRT_KF_LOCK,
1     INS$FIND_KFE,
1     INS$CVT_DIR,
1     INS$HASH;
1
1 EXTERNAL ROUTINE
1     EXE$ALLOCATE : JSB_G1_G2_3,                  ! Allocate in process space
1     EXE$ALOPAGED : JSB_G1_G2,                    ! Allocate from paged pool
1     IOC$VERIFYCHAN : JSB_0_G1,                   ! verify device channel
1     IMG$DECODE_IHD,                              ! Get and decode Image Header
1     IMG$GET_NEXT_ISD,                            ! Get and decode Image Section Descriptors
1     LIB$GET_VM,                                  ! Allocate virtual memory
1     LIB$FREE_VM,                                 ! Return virtual memory
1     MMG$GSDTRNLOG : JSB_9_G10_G11,               ! See if global section is in shared memory
1     MMG$RET_BYT_QUOTA : JSB_0,                   ! Return byte quota when sharing
1     SYS$FAO;                                     ! format ASCII data
1
1 EXTERNAL
1     ctl$gq_allocreg,                             ! Memory allocation listhead
1     ctl$gl_knownfil,                             ! Process known file listhead queues
1     EXE$GL_KNOWN_FILES : REF BBLOCK,             ! Pointer to knownfil list queues
1     EXE$GL_SYSUCB,                               ! Address of system disk unit control block
1     INS$GL_CTLMSK : BLOCK [1],                   ! Control flags
```

```
        INS$GL_KFECHAN,                              ! Channel known image file is open on.
        INS$GQ_KFERNS : $BBLOCK [DSC$C_S_BLN],       ! Result name string
        INS$GQ_KFEPRIVS : BBLOCK [8],                ! quadword privilege mask
        INS$G_KFENAM : $BBLOCK,                      ! NAM block for the filename of the known image
        INS$GL_KFEADR,                               ! Return the KFE address when it has been created or replace
        INS$L_INTRNLERR,                             ! Return internal error descriptor
        SGN$GB_KFHSHSIZ : BYTE;                      ! Number of kf list queues to put in header block

EXTERNAL LITERAL
        INS$_EXISTS,                                 ! Different version already exists
        INS$_IMGHDR,                                 ! Error reading image header
        INS$_IMGTRACED,                              ! Image linked with traceback
        INS$_INTRNLERR,                              ! INSTALL internal error
        INS$_HDRNOTRES,                              ! unable to make image header resident
        INS$_NOGBLSEC,                               ! No global sections created for shareable image
        INS$_NOHDRRES,                               ! Compatibility mode image can not be header resident
        INS$_NOSHRD,                                 ! File not shareable
        INS$_NOKFEFND,                               ! no known file entry found
        INS$_NOPAGEDYN,                              ! Not enough pagedyn
        INS$_SYSVERDIF,                              ! System version mismatch
        P1SYSVECTORS,                                ! Base of system service vectors
!       SYS$IMGSTA,                                  ! Image startup system service
        SYS$K_VERSION;                               ! Current system version value

OWN
        BLDKFDBUF : REF $BBLOCK,
        HDRBLK_BUF : REF $BBLOCK,
        IHDBUF : REF $BBLOCK,
        ISDBUF : REF $BBLOCK;

BIND
        SGN_B_KFHSHSIZ  = SGN$GB_KFHSHSIZ : BYTE;

BIND
        PROCESS_ERR_DSC = $DESCRIPTOR (' Create with /PROCESS'),
        DUPINKFD_ERR_DSC = $DESCRIPTOR (' Duplicate in KFD');

! ==========================================================================
!
!          NOTE !!
!
! The following constant is defined as a workaround for a bug in the linker.
! Because any reference to the symbol SYS$IMGSTA causes the linker to
! automatically link with /TRACEBACK and we don't want /TRACEBACK for INSTALL,
! a constant is being defined here to provide an indirect reference to
! SYS$IMGSTA instead.
!
! This constant definition is a hack and should be removed once the linker
! is fixed to allow /NOTRACEBACK for images that refer to SYS$IMGSTA. It's
! OK to have a constant because the symbol's value will never change.
!

LITERAL         SYS_IMGSTA_OFF = %X'168';       ! HARD-CODED VECTOR OFFSET

!
! ==========================================================================
1 %SBTTL 'INS$CREATE';
1
1 GLOBAL ROUTINE  INS$CREATE =
2 BEGIN
2 !+++
2 !
2 !  FUNCTIONAL DESCRIPTION:
2 !
2 !       Create a Known File entry.
2 !       If there is no listhead for the entry being created, then create one.
2 !
2 !  EXPLICIT INPUT:
2 !
2 !       none
2 !
2 !  IMPLICIT INPUT:
2 !
2 !       ins$gl_ctlmsk   =       INSTALL's control flags dictating which operation to perform
2 !       INS$GL_KFECHAN  =       Channel on which the known file image is open
2 !       INS$GQ_KFEPRIVS =       Address of quadword containing privilege mask for KFE
2 !       INS$G_KFENAM    =       Name Block to get the dir, nam and typ strings for the KFE
2 !       INS$GQ_KFERNS   =       Result Name String for error messages
2 !
2 !  IMPLICIT OUTPUT:
2 !
2 !       INS$GL_KFEADR   =       Address of KFE, may also have low bit set
2 !
2 !  ROUTINE VALUE:
2 !
2 !    R0 = return status, low bit set for success, else error status
2 !
2 !---
2
2 LOCAL
2     ONE_BLOCK,
2     STATUS;
2
2 !
2 ! Allocate buffers if needed
2 !
2 ONE_BLOCK = 512;
2 IF .HDRBLK_BUF EQL 0
2     THEN EXECUTE(LIB$GET_VM(ONE_BLOCK,HDRBLK_BUF));
2 IF .IHDBUF EQL 0
2     THEN EXECUTE(LIB$GET_VM(ONE_BLOCK,IHDBUF));
2 IF .ISDBUF EQL 0
2     THEN EXECUTE(LIB$GET_VM(ONE_BLOCK,ISDBUF));
2 IF .BLDKFDBUF EQL 0
2     THEN EXECUTE(LIB$GET_VM(%REF(KFD$C_LENGTH+NAM$C_MAXRSS),BLDKFDBUF));
2
2 STATUS = INS$EXECUTE_IN_KRNL_WITH_W_LOCK (INS_CREATE, 0);
2
2 IF .INS$GL_CTLMSK [INS$V_NOGBLSEC]
2 THEN
2     SIGNAL (INS$_NOGBLSEC,1,INS$GQ_KFERNS);
2
2 IF .INS$GL_CTLMSK [INS$V_NOHDRRES]
```

```
:  339          0712  2 THEN
:  340          0713  2      SIGNAL (INS$_NOHDRRES,1,INS$GQ_KFERNS);
:  341          0714  2
:  342          0715  2 RETURN .STATUS;
:  343          0716  1 END;              ! Global routine INS$CREATE


                                                                     .TITLE  INSCREATE
                                                                     .IDENT  \V04-000\

                                                                     .PSECT  $PLIT$,NOWRT,NOEXE,2

50  2F  20  68  74  69  77  20  65  74  61  65  72  43  20  00000 P.AAB:  .ASCII  \ Create with /PROCESS\
                                    53  53  45  43  4F  52  0000F
                                                            00015          .BLKB   3
                                                 00000015   00018 P.AAA:  .LONG   21
                                                 00000000'  0001C          .ADDRESS P.AAB
4B  20  6E  69  20  65  74  61  63  69  6C  70  75  44  20  00020 P.AAD:  .ASCII  \ Duplicate in KFD\
                                                    44  46  0002F
                                                            00031          .BLKB   3
                                                 00000011   00034 P.AAC:  .LONG   17
                                                 00000000'  00038          .ADDRESS P.AAD

                                                                           .PSECT  $OWN$,NOEXE,2

                                                            00000 BLDKFDBUF:
                                                                           .BLKB   4
                                                            00004 HDRBLK_BUF:
                                                                           .BLKB   4
                                                            00008 IHDBUF:  .BLKB   4
                                                            0000C ISDBUF:  .BLKB   4

                                                                  PROCESS_ERR_DSC=      P.AAA
                                                                  DUPINKFD_ERR_DSC=     P.AAC
                                                                           .EXTRN  INS$EXECUTE_IN_KRNL_WITH_W_LOCK
                                                                           .EXTRN  INS$CNVRT_KF_LOCK
                                                                           .EXTRN  INS$FIND_KFE, INS$CVT_DIR
                                                                           .EXTRN  INS$HASH, EXE$ALLOCATE
                                                                           .EXTRN  EXE$ALOPAGED, IOC$VERIFYCHAN
                                                                           .EXTRN  IMG$DECODE_IHD, IMG$GET_NEXT_ISD
                                                                           .EXTRN  LIB$GET_VM, LIB$FREE_VM
                                                                           .EXTRN  MMG$GSDTRNLOG, MMG$RET_BYT_QUOTA
                                                                           .EXTRN  SYS$FAO, CTL$GQ_ALLOCREG
                                                                           .EXTRN  CTL$GL_KNOWNFIL
                                                                           .EXTRN  EXE$GL_KNOWN_FILES
                                                                           .EXTRN  EXE$GL_SYSUCB, INS$GL_CTLMSK
                                                                           .EXTRN  INS$GL_KFECHAN, INS$GQ_KFERNS
                                                                           .EXTRN  INS$GQ_KFEPRIVS
                                                                           .EXTRN  INS$G_KFENAM, INS$GL_KFEADR
                                                                           .EXTRN  INS$L_INTRNLERR
                                                                           .EXTRN  SGN$GB_KFHSHSIZ
                                                                           .EXTRN  INS$_EXISTS, INS$_IMGHDR
                                                                           .EXTRN  INS$_IMGTRACED, INS$_INTRNLERR
                                                                           .EXTRN  INS$_HDRNOTRES, INS$_NOGBLSEC
                                                                           .EXTRN  INS$_NOHDRRES, INS$_NOSHRD
                                                                           .EXTRN  INS$_NOKFEFND, INS$_NOPAGEDYN
                                                                           .EXTRN  INS$_SYSVERDIF, P1SYSVECTORS
```

```
                                                                          .EXTRN  SYS$K_VERSION

                                                                          .PSECT  $CODE$,NOWRT,2

                                                      007C 00000          .ENTRY  INS$CREATE, Save R2,R3,R4,R5,R6                      ; 0657
                           56 00000000G    00    9E 00002          MOVAB   LIB$SIGNAL, R6                                       ;
                           55 00000000G    00    9E 00009          MOVAB   INS$GQ_KFERNS, R5                                    ;
                           54      0000'   CF    9E 00010          MOVAB   HDRBLK_BUF, R4                                       ;
                           53 00000000G    00    9E 00015          MOVAB   LIB$GET_VM, R3                                       ;
                           5E              08    C2 0001C          SUBL2   #8, SP                                               ;
                04         AE      0200    8F    3C 0001F          MOVZWL  #512, ONE_BLOCK                                      ; 0695
                                           64    D5 00025          TSTL    HDRBLK_BUF                                           ; 0696
                                           0B    12 00027          BNEQ    1$                                                   ;
                                           54    DD 00029          PUSHL   R4                                                   ; 0697
                                   08      AE    9F 0002B          PUSHAB  ONE_BLOCK                                            ;
                           63              02    FB 0002E          CALLS   #2, LIB$GET_VM                                       ;
                           76              50    E9 00031          BLBC    STATUS, 7$                                           ;
                                   04      A4    D5 00034   1$:    TSTL    IHDBUF                                               ; 0698
                                           0C    12 00037          BNEQ    2$                                                   ;
                                   04      A4    9F 00039          PUSHAB  IHDBUF                                               ; 0699
                                   08      AE    9F 0003C          PUSHAB  ONE_BLOCK                                            ;
                           63              02    FB 0003F          CALLS   #2, LIB$GET_VM                                       ;
                           65              50    E9 00042          BLBC    STATUS, 7$                                           ;
                                   08      A4    D5 00045   2$:    TSTL    ISDBUF                                               ; 0700
                                           0C    12 00048          BNEQ    3$                                                   ;
                                   08      A4    9F 0004A          PUSHAB  ISDBUF                                               ; 0701
                                   08      AE    9F 0004D          PUSHAB  ONE_BLOCK                                            ;
                           63              02    FB 00050          CALLS   #2, LIB$GET_VM                                       ;
                           54              50    E9 00053          BLBC    STATUS, 7$                                           ;
                                   FC      A4    D5 00056   3$:    TSTL    BLDKFDBUF                                            ; 0702
                                           12    12 00059          BNEQ    4$                                                   ;
                                   FC      A4    9F 0005B          PUSHAB  BLDKFDBUF                                            ; 0703
                04         AE      0110    8F    3C 0005E          MOVZWL  #272, 4(SP)                                          ;
                                   04      AE    9F 00064          PUSHAB  4(SP)                                                ;
                           63              02    FB 00067          CALLS   #2, LIB$GET_VM                                       ;
                           3D              50    E9 0006A          BLBC    STATUS, 7$                                           ;
                                           7E    D4 0006D   4$:    CLRL    -(SP)                                                ; 0705
                                   0000V   CF    9F 0006F          PUSHAB  INS_CREATE                                           ;
                   00000000G       00      02    FB 00073          CALLS   #2, INS$EXECUTE_IN_KRNL_WITH_W_LOCK                  ;
                           52              50    D0 0007A          MOVL    R0, STATUS                                           ;
                OD 00000000G       00      06    E1 0007D          BBC     #6, INS$GL_CTLMSK+2, 5$                              ; 0707
                                           55    DD 00085          PUSHL   R5                                                   ; 0709
                                           01    DD 00087          PUSHL   #1                                                   ;
                              00000000G    8F    DD 00089          PUSHL   #INS$_NOGBLSEC                                       ;
                           66              03    FB 0008F          CALLS   #3, LIB$SIGNAL                                       ;
                              00000000G    00    95 00092   5$:    TSTB    INS$GL_CTLMSK+2                                      ; 0711
                                           0D    18 00098          BGEQ    6$                                                   ;
                                           55    DD 0009A          PUSHL   R5                                                   ; 0713
                                           01    DD 0009C          PUSHL   #1                                                   ;
                              00000000G    8F    DD 0009E          PUSHL   #INS$_NOHDRRES                                       ;
                           66              03    FB 000A4          CALLS   #3, LIB$SIGNAL                                       ;
                           50              52    D0 000A7   6$:    MOVL    STATUS, R0                                           ; 0715
                                                 04 000AA   7$:    RET                                                          ; 0716

; Routine Size:  171 bytes,    Routine Base:  $CODE$ + 0000
```

```
INSCREATE                                               N 13
V04-000          INS$CREATE                             16-Sep-1984 01:49:49    VAX-11 Bliss-32 V4.0-742               Page   9
                                                        14-Sep-1984 12:35:36    [INSTAL.SRC]INSCREATE.B32;1                   (3)

;  344           0717  1
```

```
1 %SBTTL 'INS_CREATE';
1
1 GLOBAL ROUTINE  INS_CREATE =
2 BEGIN
2 !+++
2 !
2 !  FUNCTIONAL DESCRIPTION:
2 !
2 !       Create a Known File entry.
2 !       If there is no listhead for the entry being created, then create one.
2 !
2 !  EXPLICIT INPUT:
2 !
2 !       none
2 !
2 !  IMPLICIT INPUT:
2 !
2 !       ins$gl_ctlmsk    =       INSTALL's control flags dictating which operation to perform
2 !       INS$GL_KFECHAN   =       Channel on which the known file image is open
2 !       INS$GQ_KFEPRIVS  =       Address of quadword containing privilege mask for KFE
2 !       INS$G_KFENAM     =       Name Block to get the dir, nam and typ strings for the KFE
2 !
2 !  IMPLICIT OUTPUT:
2 !
2 !       INS$GL_KFEADR    =       Address of KFE, may also have low bit set
2 !
2 !  ROUTINE VALUE:
2 !
2 !    R0 = return status, low bit set for success, else error status
2 !
2 !---
2
2 LOCAL
2     KFD : REF BBLOCK,
2     KFD_INSERT_ADR,
2     HASH_INDEX,
2     KFE : REF BBLOCK,
2     LENGTH,
2     STATUS;
2 !
2 !   Set up initial global-section-created flag for shareable image installation.
2 !
2 INS$GL_CTLMSK [INS$V_NOGBLSEC] = FALSE;          ! Assume that /share will result in global section creation
2
2 !
2 !   Set up initial resident header created flag.
2 !
2 INS$GL_CTLMSK [INS$V_NOHDRRES] = FALSE;          ! Assume that /header is OK
2
2 !
2 !   Compute which hash table bucket Known File Entry should go into.
2 !
2 HASH_INDEX = INS$HASH (.INS$G_KFENAM [NAM$B_NAME], .INS$G_KFENAM [NAM$L_NAME],
2                         .SGN_B_KFHSHSIZ );
2 !
2 !   Check for another version of this image already installed, that is, a file name
2 !   that is equal and from the same device, directory and with the same file type
```

```
;  403          0775  2 !   as the one we are currently trying to install.
;  404          0776  2 !
;  405          0777  2 STATUS = INS$FIND_KFE (.HASH_INDEX, INS$G_KFENAM);
;  406          0778  2 IF .STATUS NEQ 0
;  407          0779  2 THEN
;  408          0780  2     RETURN INS$_EXISTS;
;  409          0781  2
;  410          0782  2 !
;  411          0783  2 !   Check if the Known File Device, Directory, Type (KFD) block exists.
;  412          0784  2 !   If it doesn't, record where it should be inserted when it is created.
;  413          0785  2 !
;  414          0786  2 KFD = FIND_KFD (INS$G_KFENAM, KFD_INSERT_ADR);
;  415          0787  2
;  416          0788  2 STATUS = CREATE (.HASH_INDEX, .KFD, .KFD_INSERT_ADR);
;  417          0789  2
;  418          0790  2 RETURN .STATUS;
;  419          0791  1 END;            ! Global routine INS_CREATE



                                         001C 00000         .ENTRY  INS_CREATE, Save R2,R3,R4                        ; 0720
                        54 00000000G  00   9E 00002         MOVAB   INS$G_KFENAM, R4                                 ;
                        5E            04   C2 00009         SUBL2   #4, SP                                           ;
             00000000G  00         C0  8F   8A 0000C         BICB2   #192, INS$GL_CTLMSK+2                            ; 0765
                        7E 00000000G  00   9A 00014         MOVZBL  SGN_B_KFHSHSIZ, -(SP)                            ; 0771
                                   4C  A4   DD 0001B         PUSHL   INS$G_KFENAM+76                                  ; 0770
                        7E         3B  A4   9A 0001E         MOVZBL  INS$G_KFENAM+59, -(SP)                           ;
             00000000G  00             03   FB 00022         CALLS   #3, INS$HASH                                     ;
                        53             50   D0 00029         MOVL    R0, HASH_INDEX                                   ;
                                       18   BB 0002C         PUSHR   #^M<R3,R4>                                       ; 0777
             00000000G  00             02   FB 0002E         CALLS   #2, INS$FIND_KFE                                 ;
                        52             50   D0 00035         MOVL    R0, STATUS                                       ;
                                       08   13 00038         BEQL    1$                                               ; 0778
                        50 00000000G  8F   D0 0003A         MOVL    #INS$_EXISTS, R0                                 ; 0780
                                            04 00041         RET                                                      ;
                                 4010  8F   BB 00042 1$:     PUSHR   #^M<R4,SP>                                       ; 0786
                  0000V  CF             02   FB 00046         CALLS   #2, FIND_KFD                                     ;
                                       6E   DD 0004B         PUSHL   KFD_INSERT_ADR                                   ; 0788
                                       50   DD 0004D         PUSHL   KFD                                              ;
                                       53   DD 0004F         PUSHL   HASH_INDEX                                       ;
                  0000V  CF             03   FB 00051         CALLS   #3, CREATE                                       ;
                        52             50   D0 00056         MOVL    R0, STATUS                                       ;
                                            04 00059         RET                                                      ; 0791

; Routine Size:  90 bytes,    Routine Base:  $CODE$ + 00AB


;  420          0792  1
```

```
1 %SBTTL 'create';
1
1 ROUTINE CREATE (HASH_INDEX, KFD, KFD_INSERT_ADR ) =
2 BEGIN
2 !++
2 !
2 !   FUNCTIONAL DESCRIPTION:
2 !
2 !        Create a Known File entry.
2 !        If there is no listhead for the entry being created, then create one.
2 !        Execute in Kernel mode
2 !
2 !   EXPLICIT INPUT:
2 !
2 !        HASH_INDEX      Index of Hash bucket the new KFE should be inserted in
2 !        KFD             Device, Directory, Type block if it exists.
2 !        KFD_INSERT_ADR  Address to insert a KFD if one does not exist and
2 !                        much be built
2 !
2 !   IMPLICIT INPUT:
2 !
2 !        ins$gl_ctlmsk   =       INSTALL's control flags dictating which operation to perform
2 !        INS$GL_KFECHAN  =       Channel on which the known file image is open
2 !        INS$GQ_KFEPRIVS =       Address of quadword containing privilege mask for KFE
2 !        INS$G_KFENAM    =       Name Block to get the dir, nam and typ strings for the KFE
2 !
2 !   IMPLICIT OUTPUT:
2 !
2 !        INS$GL_KFEADR   =       Address of KFE, may also have low bit set
2 !
2 !   ROUTINE VALUE:
2 !
2 !     R0 = return status, low bit set for success, else error status
2 !
2 !--
2 LOCAL
2     CCB : REF BBLOCK,
2     WCB : REF BBLOCK,
2     KFE : REF BBLOCK,
2     BLD_KFE_BUF : $BBLOCK [KFE$C_LENGTH + 39],  ! Size of entry plus max size of NAM block file name field
2     LENGTH,
2     HDR_VERSION,
2     ALIAS : WORD,
2     OFFSET,
2     VBN,
2     STATUS;
2 MAP
2     KFD : REF BBLOCK;
2
2
2 IF .INS$GL_CTLMSK [INS$V_PROCESS]
2 THEN
3     BEGIN
3     INS$L_INTRNLERR = PROCESS_ERR_DSC;
3     RETURN INS$_INTRNLERR;                     ! replace with call to ins$p1permanent ();
2     END;
2
```

```
2 !
2 !     Build a Known File Entry (KFE) for later insertion into hash bucket list
2 !
2 LENGTH = KFE$C_LENGTH + .INS$G_KFENAM [NAM$B_NAME];
2 KFE = BLD_KFE_BUF;                     ! Point to buffer on stack, copy to paged pool when its time to enqu
2 CH$FILL (0, .LENGTH, .KFE);             ! zero the KFE
2
2 KFE [KFE$W_SIZE] = .LENGTH;
2 KFE [KFE$B_TYPE] = DYN$C_KFE;
2 KFE [KFE$B_HSHIDX] = .HASH_INDEX;
2
2 !
2 !     Store the file name in the KFE.  There will be a pointer to the
2 !     device, directory and type which will be stored in a KFD block.
2 !
2 KFE [KFE$B_FILNAMLEN] = .INS$G_KFENAM [NAM$B_NAME];
2 CH$MOVE (.INS$G_KFENAM [NAM$B_NAME], .INS$G_KFENAM [NAM$L_NAME],
2                KFE [KFE$T_FILNAM]);
2
2 KFE [KFE$V_HDRRES] = .INS$GL_CTLMSK [INS$V_HDRRES];
2 KFE [KFE$V_SHARED] = .INS$GL_CTLMSK [INS$V_SHARED];
2 KFE [KFE$V_PROTECT] = .INS$GL_CTLMSK [INS$V_PROTECT];
2 KFE [KFE$V_OPEN] = .INS$GL_CTLMSK [INS$V_OPEN];
2 KFE [KFE$V_NOPURGE] = .INS$GL_CTLMSK [INS$V_NOPURGE];
2 KFE [KFE$V_ACCOUNT] = .INS$GL_CTLMSK [INS$V_ACCOUNT];
2 KFE [KFE$V_EXEONLY] = .INS$GL_CTLMSK [INS$V_EXEONLY];
2
2 IF .INS$GL_CTLMSK [INS$V_SHARED]
2 THEN
2     KFE [KFE$V_WRITEABLE] = .INS$GL_CTLMSK [INS$V_WRITABLE];
2
2 IF .INS$GL_CTLMSK [INS$V_SHARED] OR              ! /SHARE or /HEAD implies /OPEN
2     .INS$GL_CTLMSK [INS$V_HDRRES]
2 THEN
2     KFE [KFE$V_OPEN] = TRUE;
2
2 STATUS = VERIFY_CHANNEL (.INS$GL_KFECHAN, CCB); ! Obtain the CCB
2 IF NOT .STATUS THEN RETURN .STATUS;
2 IF NOT .CCB [CCB$L_UCB] EQL .EXE$GL_SYSUCB      ! If this is not the system device
2 THEN
2     IF .INS$GL_CTLMSK [INS$V_PRIV]             ! Then a privileged image must remain open
2     THEN                                       ! to keep a transaction against the volume
2         KFE [KFE$V_OPEN] = TRUE;
2
2 IF .INS$GL_CTLMSK [INS$V_PRIV]
2 THEN
3     BEGIN
3     KFE [KFE$V_PROCPRIV] = TRUE;                ! If installed /PRIV
3     CH$MOVE (8, INS$GQ_KFEPRIVS, KFE [KFE$Q_PROCPRIV]); ! copy in the privilege mask
2     END;
2
2 !
2 !     Check if the Known File Device Directory, Type (KFD) block exists.
2 !     If it doesn't create it for later insertion in KFD list
2 !
2 IF .KFD EQL 0
2 THEN
```

```
;   536         0907  2         BUILD_KFD (INS$G_KFENAM,.BLDKFDBUF)
;   537         0908  2     ELSE
;   538         0909  2         KFE [KFE$L_KFD] = .KFD;                         ! KFD exists and is in place
;   539         0910  2
;   540         0911  2     !
;   541         0912  2     !       The image header is opened for a number of reasons.
;   542         0913  2     !
;   543         0914  3     IF      (.KFE [KFE$V_PROCPRIV]
;   544         0915  3         OR .KFE [KFE$V_EXEONLY]
;   545         0916  3         OR .KFE [KFE$V_OPEN])
;   546         0917  2     THEN
;   547         0918  3         BEGIN
;   548         0919  3         !
;   549         0920  3         !   Read the image header.
;   550         0921  3         !
;   551         0922  3         CH$FILL (0, 512, .HDRBLK_BUF);
;   552         0923  3         CH$FILL (0, 512, .IHDBUF);
;   553         0924  3         STATUS = IMG$DECODE_IHD (.INS$GL_KFECHAN, .HDRBLK_BUF, .IHDBUF,
;   554         0925  3                         VBN, OFFSET, HDR_VERSION, ALIAS);
;   555         0926  3         IF NOT .STATUS THEN RETURN .STATUS;
;   556         0927  2         END;
;   557         0928  2
;   558         0929  2     !
;   559         0930  2     !       Verify that the image transfer array doesn't contain SYS$IMGSTA for
;   560         0931  2     !       images installed with privilege or as execute_only images.
;   561         0932  2     !
;   562         0933  2     IF .KFE [KFE$V_PROCPRIV] OR .KFE [KFE$V_EXEONLY]
;   563         0934  2     THEN
;   564         0935  3         BEGIN
;   565         0936  3         LOCAL
;   566         0937  3             ACTIVOFF : BBLOCK [IHA$C_LENGTH],
;   567         0938  3             TFR1;
;   568         0939  3
;   569         0940  3         ACTIVOFF = .IHDBUF + .IHDBUF [IHD$W_ACTIVOFF];
;   570         0941  3         TFR1 = .(.ACTIVOFF [IHA$L_TFRADR1]);        ! Get first image transfer address
;   571         0942  4         IF (.TFR1 EQL (P1SYSVECTORS + SYS_IMGSTA_OFF))
;   572         0943  3             OR
;   573         0944  4             ((.TFR1 - %X'80000000') EQL SYS_IMGSTA_OFF)
;   574         0945  3         THEN
;   575         0946  3             RETURN INS$_IMGTRACED;
;   576         0947  2         END;
;   577         0948  2
;   578         0949  2     IF NOT .KFE [KFE$V_OPEN]
;   579         0950  2     THEN
;   580         0951  2         CH$MOVE (8, INS$G_KFENAM [NAM$W_FID], KFE [KFE$W_FID])
;   581         0952  2     !
;   582         0953  2     !       Explicit or implicit /OPEN. If /HEAD then store the image header.
;   583         0954  2     !       If /SHARE, then process the ISDs and build global sections.
;   584         0955  2     !
;   585         0956  2     ELSE
;   586         0957  3         BEGIN
;   587         0958  3         LOCAL
;   588         0959  3             BLDHDR_LEN,
;   589         0960  3             CRESECFLG,                          ! Mask of create section options
;   590         0961  3             GBLSECNAM_DSC : BBLOCK [DSC$C_S_BLN],   ! Address of descriptor of global section name
;   591         0962  3             GBLSECNAM : BBLOCK [INS$C_GBLNAMLEN],
;   592         0963  3             BLDHDR : REF BBLOCK,
3          BLDHDR_SIZ;
3
3      !
3      !   Do some image type specific processing.
3      !
3      IF
4          (.ALIAS EQL IHD$C_RSX)
3          OR
4          (.ALIAS EQL IHD$C_BPA)
3          OR
4          (.ALIAS EQL IHD$C_ALIAS)
3      THEN
3
3          !
3          !   If it's not a native mode image, then set the COMPAT flag,
3          !   disallow a resident header, and store the AME type code.
3          !
4          BEGIN
4          KFE [KFE$V_COMPATMOD] = TRUE;
4          IF .INS$GL_CTLMSK [INS$V_HDRRES]
4          THEN
5              BEGIN
5              INS$GL_CTLMSK [INS$V_HDRRES] = FALSE;
5              KFE [KFE$V_HDRRES] = FALSE;
5              INS$GL_CTLMSK [INS$V_NOHDRRES] = TRUE;
4              END;
4          KFE [KFE$W_AMECOD] = .ALIAS;            ! Store which type of AME
4          END
3      ELSE
3          !
3          !   If it's a native mode image, determine if it's shareable. Also,
3          !   perform special checks on the header if it's going to be resident.
3          !
4          BEGIN
4          BIND
4              MINORID_DIGIT = IHDBUF [IHD$W_MINORID] : VECTOR [2,BYTE];
4
4          LITERAL
4              MINOR_ID_TENS = IHD$K_MINORID AND %X'FF',
4              MINOR_ID_ONES = IHD$K_MINORID ^ -8;
4
4          !
4          !   Determine if this image is shareable.
4          !
4          KFE [KFE$V_LIM] = (.IHDBUF [IHD$B_IMGTYPE] EQL IHD$K_LIM);
4
4          IF .KFE [KFE$V_HDRRES]
4          THEN
4              !
4              !   The major ID in the image header must be identically equal to
4              !   the constant IHD$K_MAJORID. The minor ID in the image header
4              !   must be LEQU the constant IHD$K_MINORID. Both IDs are stored
4              !   as ASCII strings.
4              !
5              BEGIN
6              IF (.IHDBUF [IHD$W_MAJORID] NEQU IHD$K_MAJORID)
5              THEN RETURN SS$_BADIMGHDR;
```

```
                IF (
                    (.MINORID_DIGIT [0] GTRU MINOR_ID_TENS)
                    OR
                    (
                     (.MINORID_DIGIT [0] EQLU MINOR_ID_TENS)
                     AND
                     (.MINORID_DIGIT [1] GTRU MINOR_ID_ONES)
                    )
                   )
                THEN RETURN SS$_BADIMGHDR;

                !
                !   If the image was linked against a SYS.STB for other than
                !   the current system, then don't install it.
                !
                IF (.IHDBUF [IHD$L_SYSVER] NEQU 0)
                THEN
                    IF (.IHDBUF [IHD$L_SYSVER] NEQU SYS$K_VERSION)
                    THEN RETURN INS$_SYSVERDIF;
                END;
            END;

        !
        !   Perform some initialization of the Create and Map Section parameters
        !
        IF .INS$GL_CTLMSK [INS$V_SHARED]    ! /SHARE
        THEN
            BEGIN
            LOCAL
                IS_SHRMEM;

            !
            !   Init global section name
            !
            CH$FILL (0, INS$C_GBLNAMLEN, GBLSECNAM);
            GBLSECNAM_DSC = 0;
            GBLSECNAM_DSC [DSC$A_POINTER] = GBLSECNAM;
            INS$BLD_GBLSECNAM (GBLSECNAM_DSC);      ! Build the global section name, FILENAM_nnn
```

```
IF .KFE [KFE$V_COMPATMOD]
THEN
    BEGIN
    IF .ALIAS NEQ IHD$C_RSX
    THEN
        BEGIN
        IF .INS$GL_CTLMSK [INS$V_SHARED]
        THEN
            BEGIN
            INS$GL_CTLMSK [INS$V_SHARED] = FALSE;
            KFE [KFE$V_SHARED] = FALSE;
            !! Perhaps it is now implicitly OPEN
            RETURN INS$_NOSHRD;
            END;
        END
    ELSE                    ! RSX AME
        BEGIN
        LOCAL
            N_DSC,      ! number of descriptors in RSX image header
            PAGCNT,
            VBN;

        !
        !   Would a global section that might exist for this image
        !   be in shared memory?
        !
        STATUS = CHECK_SHMIDENT (GBLSECNAM_DSC, IS_SHRMEM);
        IF NOT .STATUS THEN RETURN .STATUS;
        KFE [KFE$V_SHMIDENT] = .IS_SHRMEM;      ! Record SHM state

        !
        !   Set up the match control and IDENT for global sections.
        !   Extract the flags word from the Compatibility mode
        !   image header and see if the TS$NHD bit is set.
        !   If the No_header bit is not set, there is a header,
        !   so use the date in the header, else use 0.
        !
        KFE [KFE$B_MATCHCTL] = ISD$K_MATEQU;

        IF (.(.IHDBUF + $BYTEOFFSET(L$BFLG) ) <0,16> AND TS$NHD) EQL 0
        THEN
            KFE [KFE$L_IDENT] = .(.IHDBUF + $BYTEOFFSET (L$BDAT) + 2)
        ELSE
            KFE [KFE$L_IDENT] = 0;

        !
        !   Obtain VBN and Page count
        !
        IF .(.IHDBUF + $BYTEOFFSET (L$BSYS) ) <0,8> NEQ 4
        THEN                        ! RSX-11M Task, there are 7 descriptors
            N_DSC = 0
        ELSE                        ! Not an RSX-11M task so allow for 8 more descriptors
            N_DSC = (8 * ($BYTEOFFSET (L$BLIB) - $BYTEOFFSET (L$BPAR)));

        IF (.(.IHDBUF + $BYTEOFFSET(L$BFLG) ) <0,16> AND TS$NHD) EQL 0
        THEN
            !
```

```
                !   There is a header, so figure out which type so we can
                !   skip past the correct number of descriptors to get the
                !   VBN and PAGE COUNT.
                !
                BEGIN
                VBN = .(.IHDBUF + $BYTEOFFSET (L$BROB) + .N_DSC ) <0,16>;
                PAGCNT = .(.IHDBUF + $BYTEOFFSET (L$BROL) + .N_DSC ) <0,16>;        ! Number of 64 byte
                END
            ELSE
                BEGIN       ! There is no header, treat as a Library Common
                VBN = .(.IHDBUF + $BYTEOFFSET (L$BHRB) + .N_DSC ) <0,16> + 1;
                PAGCNT = .(.IHDBUF + $BYTEOFFSET (L$BLDZ) ) <0,16>;                 ! Number of 64 byte
                END;

            !
            ! Check PAGCNT for zero. If zero, then this task was not built with a shareable
            ! section. Don't continue here. Just report the fact that no global sections
            ! were created.
            !
            IF .PAGCNT EQL 0
            THEN
                BEGIN
                INS$GL_CTLMSK [INS$V_NOGBLSEC] = TRUE;
                INS$GL_CTLMSK [INS$V_SHARED] = FALSE;
                KFE [KFE$V_SHARED] = FALSE;
                KFE [KFE$V_SHMIDENT] = FALSE;
                END
            ELSE
                BEGIN
                PAGCNT = .PAGCNT + 7;                   ! Round up to next 512 bytes
                PAGCNT = .PAGCNT / 8;                   ! Divide to get page count
                CRESECFLG = SEC$M_GBL OR SEC$M_SYSGBL OR
                            SEC$M_PERM;                 ! Create a permanent system global section

                IF .INS$GL_CTLMSK [INS$V_WRITABLE]
                THEN
                    CRESECFLG = .CRESECFLG OR SEC$M_WRT;

                !
                !   Create Global section
                !

                STATUS = $CRMPSC (
                    INADR = 0,                          ! Create but don't map
                    ACMODE = PSL$C_USER,                ! Access mode
                    FLAGS = .CRESECFLG,                 ! Mask of create options
                    GSDNAM = GBLSECNAM_DSC,             ! Address of descriptor of global section name
                    IDENT = KFE [KFE$B_MATCHCTL],       ! Address of quadword containing ident
                    CHAN = .INS$GL_KFECHAN,             ! Channel file is open on
                    PAGCNT = .PAGCNT,                   ! Number of pages in section
                    VBN = .VBN                          ! Virtual block number
                                  );
                IF .STATUS
                THEN
                    KFE [KFE$W_GBLSECCNT] = 1
                ELSE
                    RETURN .STATUS;                     ! Report global section creation failure
```

```
;   805              1175  6                     END;                  ! Compat with RSX AME
;   806              1176  5                 END;
;   807              1177  5             END                   ! Shared COMPAT
```

```
        ELSE
            !
            !   Shared Native mode image
            !
            BEGIN
            CRESECFLG = 0;                          ! Mask of create options

            !
            !   Determine the Ident and Match control to use if global sections
            !   are to be created.  Store in quadword GBLSEC_MATCH_IDENT with
            !   Ident in second longword.
            !
            KFE [KFE$B_MATCHCTL] = ISD$K_MATEQU;                        ! Default, assuming not shareable im
            KFE [KFE$L_IDENT] = .IHDBUF [IHD$L_IDENT];                  ! Use Header ident as default ident
            IF .KFE [KFE$V_LIM]                                         ! Is it a shareable image?
            THEN
                BEGIN
                IF .IHDBUF [IHD$V_MATCHCTL] EQL 0
                THEN
                    KFE [KFE$L_IDENT] = 0;                              ! Match always
                KFE [KFE$B_MATCHCTL] = .IHDBUF [IHD$V_MATCHCTL];
                END;

            !
            !   Check if image is in shared memory
            !   This will affect the ident and match control
            !
            STATUS = CHECK_SHMIDENT (GBLSECNAM_DSC, IS_SHRMEM);
            IF NOT .STATUS THEN RETURN .STATUS;
            KFE [KFE$V_SHMIDENT] = .IS_SHRMEM;
            IF .IS_SHRMEM AND NOT .KFE [KFE$V_LIM]
            THEN
                BEGIN
                !
                !   If its been patched, use patch date as ident,
                !   else use date in Image Header Ident
                !
                KFE [KFE$L_IDENT] =
                    (IF .IHDBUF [IHD$W_PATCHOFF] EQL 0
                    THEN
                        BEGIN
                        BIND
                            IHI = .IHDBUF + .IHDBUF [IHD$W_IMGIDOFF] : BBLOCK;
                        .(IHI [IHI$Q_LINKTIME] + 2)
                        END
                    ELSE
                        BEGIN
                        BIND
                            IHP = .IHDBUF + .IHDBUF [IHD$W_PATCHOFF] : BBLOCK;
                        .(IHP [IHP$Q_PATDATE] + 2)
                        END
                    );
                KFE [KFE$B_MATCHCTL] = ISD$K_MATEQU;
                END;

            END;        ! Initialize for SHARED not COMPAT
```

```
;  866            1235  3          END;    ! Initialize for /SHARE
;  867            1236  3
```

```
!+++
!
!    Save header if its to be made resident
!
!+++

IF .KFE [KFE$V_HDRRES]
THEN
    BEGIN
    BLDHDR_LEN = 512;
    EXECUTE(LIB$GET_VM (BLDHDR_LEN, BLDHDR));

    CH$FILL (0, .BLDHDR_LEN, .BLDHDR);      ! zero the buffer

    CH$MOVE (.IHDBUF [IHD$W_SIZE], .IHDBUF, .BLDHDR);
    BLDHDR_SIZ = .IHDBUF [IHD$W_SIZE];
    END;
```

```
;   888          1255  3       IF NOT .KFE [KFE$V_COMPATMOD]
;   889          1256  3       THEN
;   890          1257  4           BEGIN
;   891          1258  4
;   892          1259  4           !+++
;   893          1260  4           !
;   894          1261  4           !    ISD processing loop
;   895          1262  4           !
;   896          1263  4           !+++
;   897          1264  4
;   898          1265  4           CH$FILL (0, 512, .ISDBUF);
;   899          1266  5           WHILE (STATUS = IMG$GET_NEXT_ISD (.INS$GL_KFECHAN, .HDRBLK_BUF, .IHDBUF,
;   900          1267  4                   VBN, OFFSET, .ISDBUF, .HDR_VERSION) ) DO
;   901          1268  5               BEGIN
;   902          1269  5
;   903          1270  5               IF .KFE [KFE$V_HDRRES]
;   904          1271  5               THEN
;   905          1272  5                   !
;   906          1273  5                   !    Concatenate this ISD onto stored header
;   907          1274  5                   !
;   908          1275  6                   BEGIN
;   909          1276  6                   IF .BLDHDR_SIZ + .ISDBUF [ISD$W_SIZE] GTR .BLDHDR_LEN
;   910          1277  6                   THEN
;   911          1278  7                       BEGIN
;   912          1279  7                       LOCAL
;   913          1280  7                           NEW_BLDHDR,
;   914          1281  7                           NEW_BLDHDR_LEN;
;   915          1282  7
;   916          1283  7                       NEW_BLDHDR_LEN = 2 * .BLDHDR_LEN;
;   917          1284  7                       EXECUTE(LIB$GET_VM (NEW_BLDHDR_LEN, NEW_BLDHDR));
;   918          1285  7                       CH$FILL (0, .NEW_BLDHDR_LEN, .NEW_BLDHDR);
;   919          1286  7                       CH$MOVE (.BLDHDR_SIZ, .BLDHDR, .NEW_BLDHDR);
;   920          1287  7                       EXECUTE(LIB$FREE_VM (BLDHDR_LEN, BLDHDR));
;   921          1288  7                       BLDHDR = .NEW_BLDHDR;
;   922          1289  7                       BLDHDR_LEN = .NEW_BLDHDR_LEN;
;   923          1290  6                       END;
;   924          1291  6
;   925          1292  6                   CH$MOVE (.ISDBUF [ISD$W_SIZE], .ISDBUF, (.BLDHDR + .BLDHDR_SIZ) );
;   926          1293  6                   BLDHDR_SIZ = .BLDHDR_SIZ + .ISDBUF [ISD$W_SIZE];
;   927          1294  5                   END;     ! If /HEAD then save this ISD
;   928          1295  5
```

```
  5      !
  5      !   If /SHARE then create global sections for the images private sections
  5      !
  5      IF .INS$GL_CTLMSK [INS$V_SHARED]    ! /SHARE
  5      THEN
  6          BEGIN
  6          BIND
  6              ISD = .ISDBUF : BBLOCK;
  6
  7          IF NOT (.ISD [ISD$V_GBL] OR .ISD [ISD$V_DZRO]
  7              OR .ISD [ISD$V_CRF])
  6          THEN
  7              BEGIN
  7              LOCAL
  7                  RETADR : BBLOCK [8];
  7
  7              CRESECFLG = .ISDBUF [ISD$L_FLAGS] AND ISD$M_WRT;
  7              CRESECFLG = .CRESECFLG OR SEC$M_GBL
  7                          OR SEC$M_SYSGBL OR SEC$M_PERM;  ! Create a permanent system global section
  7
  7              IF .ISDBUF [ISD$V_PROTECT] OR
  8                  (.KFE [KFE$V_PROTECT] AND NOT .ISDBUF [ISD$V_WRT])
  7              THEN
  8                  BEGIN
  8                  CRESECFLG = .CRESECFLG OR SEC$M_PROTECT;
  8                  CRESECFLG = .CRESECFLG OR PSL$C_EXEC ^ ($BITPOSITION(SEC$V_WRTMOD));
  7                  END;
  7
P 7              STATUS = $CRMPSC (
P 7                  INADR = 0,                          ! Create but don't map
P 7                  RETADR = RETADR,                    ! Create but don't map
P 7                  ACMODE = PSL$C_USER,                ! Access mode
P 7                  FLAGS = .CRESECFLG,                 ! Mask of create options
P 7                  GSDNAM = GBLSECNAM_DSC,             ! Address of descriptor of global section name
P 7                  IDENT = KFE [KFE$B_MATCHCTL],       ! Address of quadword containing ident
P 7                  RELPAG = 0,                         ! Create, don't map
P 7                  CHAN = .INS$GL_KFECHAN,             ! Channel file is open on
P 7                  PAGCNT = .ISDBUF [ISD$W_PAGCNT],    ! Number of pages in section
P 7                  VBN = .ISDBUF [ISD$L_VBN],          ! Virtual block number
P 7                  PROT = 0,                           ! Default protection mask
P 7                                                      !! want to ignore PFC if cross linker format
P 7                  PFC = .ISDBUF [ISD$B_PFC]           ! Page fault cluster size
  7                          );
  7              IF .STATUS
  7              THEN
  8                  BEGIN
  8                  INS$BLD_GBLSECNAM (GBLSECNAM_DSC);      ! Increment for the next global section name
  8                  KFE [KFE$W_GBLSECCNT] = .KFE [KFE$W_GBLSECCNT] + 1;
  8                  END
  7              ELSE
  7                  RETURN .STATUS;
  6              END;
  5          END;                          ! End of processing this ISD for /SHARE
  5
  5      CH$FILL (0, 512, .ISDBUF);
  4      END;                                      ! While getting ISD's
  4
```

```
5           IF NOT .STATUS AND (.STATUS NEQ IMG$_ENDOFHDR)
4           THEN
5               BEGIN
5               RETURN .STATUS;
4               END;
4
5           IF .INS$GL_CTLMSK [INS$V_SHARED] AND (.KFE [KFE$W_GBLSECCNT] EQLU 0)
4           THEN
5               BEGIN
5               INS$GL_CTLMSK [INS$V_NOGBLSEC] = TRUE;
5               INS$GL_CTLMSK [INS$V_SHARED] = FALSE;
5               KFE [KFE$V_SHARED] = FALSE;
5               KFE [KFE$V_SHMIDENT] = FALSE;
4               END;
4
4           IF .KFE [KFE$V_HDRRES]
4           THEN
4               !
4               !   Make the header resident
4               !
5               BEGIN
5               LOCAL
5                   KFRH : REF BBLOCK;
5
5               LENGTH = KFRH$C_LENGTH + .BLDHDR_SIZ + 4;   ! Leave longword of zeros to mark end
5               EXECUTE(ALLOC_PAGED ( .LENGTH, KFRH ));
5               CH$FILL (0, .LENGTH, .KFRH);                 ! zero the KFRH
5
5               KFRH [KFRH$W_ALIAS] = .ALIAS;
5               KFRH [KFRH$W_SIZE] = .LENGTH;
5               KFRH [KFRH$B_TYPE] = DYN$C_KFRH;
5               KFRH [KFRH$B_HDRVER] = .HDR_VERSION;
5               KFE [KFE$L_IMGHDR] = KFRH [KFRH$T_IHD];
5               CH$MOVE (.BLDHDR_SIZ, .BLDHDR, KFRH [KFRH$T_IHD]);
5               KFRH [KFRH$L_BUFEND] = KFRH [KFRH$T_IHD] + .BLDHDR_SIZ;
5               EXECUTE(LIB$FREE_VM(BLDHDR_SIZ,BLDHDR));    !Deallocate the header
4               END;
3           END;                                            ! /OPEN but not COMPAT
3
3       KFE [KFE$W_SHRCNT] = 1;                             ! Initialize shared counter (normalized on display)
3       WCB = .CCB [CCB$L_WIND];                            ! window address
3       KFE [KFE$L_WCB] = .WCB;                             ! Save window address
3       !
3       ! This call is effectively a no-op if any global sections had been created
3       !
3       MMG$RET_BYT_QUOTA (.WCB);                           ! Return byte quota since file was being opened for everyone
3       WCB [WCB$W_REFCNT] = .WCB [WCB$W_REFCNT] +1;       ! jimmy window so the shared
3                                                           !  file remains open.
2       END;
2
2 STATUS = ENTER_KFE (.KFE, .HASH_INDEX, .BLDKFDBUF, .KFD_INSERT_ADR);
2
2 RETURN .STATUS;
1 END;              ! routine CREATE
```

```
                                                                            .EXTRN   SYS$CRMPSC

                                                         0FFC 00000 CREATE: .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11      ; 0795
                                        5E        FF30  CE  9E 00002         MOVAB    -208(SP), SP                              ;
                          11 00000000G  00             01  E1 00007         BBC      #1, INS$GL_CTLMSK, 1$                     ; 0843
                             00000000G  00      0000'  CF  9E 0000F         MOVAB    PROCESS_ERR_DSC, INS$L_INTRNLERR          ; 0846
                                        50  00000000G  8F  D0 00018         MOVL     #INS$_INTRNLERR, R0                       ; 0847
                                                           04 0001F         RET                                                ;
                                        57  00000000G  00  9A 00020 1$:     MOVZBL   INS$G_KFENAM+59, R7                       ; 0853
                                        56         37  A7  9E 00027         MOVAB    55(R7), LENGTH                            ;
                                        58         70  AE  9E 0002B         MOVAB    BLD_KFE_BUF, KFE                          ; 0854
            56            00            6E             00  2C 0002F         MOVC5    #0, (SP), #0, LENGTH, (KFE)               ; 0855
                                                       68     00034                                                            ;
                                    0B  A8             56  B0 00035         MOVW     LENGTH, 8(KFE)                            ; 0857
                                    0A  A8             18  90 00039         MOVB     #24, 10(KFE)                              ; 0858
                                    0B  A8         04  AC  90 0003D         MOVB     HASH_INDEX, 11(KFE)                       ; 0859
                                    36  A8             57  90 00042         MOVB     R7, 54(KFE)                               ; 0865
                                        50  00000000G  00  D0 00046         MOVL     INS$G_KFENAM+76, R0                       ; 0866
                     37   A8            60             57  28 0004D         MOVC3    R7, (R0), 55(KFE)                         ; 0867
                                        57         10  A8  9E 00052         MOVAB    16(KFE), R7                               ; 0869
            51 00000000G  00            01             06  EF 00056         EXTZV    #6, #1, INS$GL_CTLMSK+1, R1               ;
            67            01            04             51  F0 0005F         INSV     R1, #4, #1, (R7)                          ;
            50 00000000G  00            01             01  EF 00064         EXTZV    #1, #1, INS$GL_CTLMSK+2, R0               ; 0870
            67            01            05             50  F0 0006D         INSV     R0, #5, #1, (R7)                          ;
            67            01            00  00000000G  00  F0 00072         INSV     INS$GL_CTLMSK+2, #0, #1, (R7)             ; 0871
            52 00000000G  00            01             05  EF 0007B         EXTZV    #5, #1, INS$GL_CTLMSK+1, R2               ; 0872
            67            01            03             52  F0 00084         INSV     R2, #3, #1, (R7)                          ;
            52 00000000G  00            01             03  EF 00089         EXTZV    #3, #1, INS$GL_CTLMSK+2, R2               ; 0873
      01    A7            01            00             52  F0 00092         INSV     R2, #0, #1, 1(R7)                         ;
            52 00000000G  00            01             04  EF 00098         EXTZV    #4, #1, INS$GL_CTLMSK+2, R2               ; 0874
            67            01            09             52  F0 000A1         INSV     R2, #9, #1, (R7)                          ;
            52 00000000G  00            01             05  EF 000A6         EXTZV    #5, #1, INS$GL_CTLMSK+2, R2               ; 0875
            67            01            0B             52  F0 000AF         INSV     R2, #11, #1, (R7)                         ;
                                        11             50  E9 000B4         BLBC     R0, 2$                                    ; 0877
            52 00000000G  00            01             02  EF 000B7         EXTZV    #2, #1, INS$GL_CTLMSK+2, R2               ; 0879
            67            01            0A             52  F0 000C0         INSV     R2, #10, #1, (R7)                         ;
                                        03             50  E8 000C5         BLBS     R0, 3$                                    ; 0881
                                        03             51  E9 000C8 2$:     BLBC     R1, 4$                                    ; 0882
                                        67             08  88 000CB 3$:     BISB2    #8, (R7)                                  ; 0884
                                                   04  AE  9F 000CE 4$:     PUSHAB   CCB                                       ; 0886
                                            00000000G  00  DD 000D1         PUSHL    INS$GL_KFECHAN                            ;
                                 0000V  CF             02  FB 000D7         CALLS    #2, VERIFY_CHANNEL                        ;
                                        5A             50  D0 000DC         MOVL     R0, STATUS                                ;
                                        03             5A  E8 000DF         BLBS     STATUS, 5$                                ; 0887
                                                     04D5  31 000E2         BRW      63$                                       ;
                             00000000G  00         04  BE  D1 000E5 5$:     CMPL     @CCB, EXE$GL_SYSUCB                       ; 0888
                                                       0B  13 000ED         BEQL     6$                                        ;
                                            00000000G  00  95 000EF         TSTB     INS$GL_CTLMSK+1                           ; 0890
                                                       03  18 000F5         BGEQ     6$                                        ;
                                        67             08  88 000F7         BISB2    #8, (R7)                                  ; 0892
                                            00000000G  00  95 000FA 6$:     TSTB     INS$GL_CTLMSK+1                           ; 0894
                                                       0C  18 00100         BGEQ     7$                                        ;
                                        67             04  88 00102         BISB2    #4, (R7)                                  ; 0897
                     20   A8 00000000G  00             08  28 00105         MOVC3    #8, INS$GQ_KFEPRIVS, 32(KFE)              ; 0898
                                                   08  AC  D5 0010E 7$:     TSTL     KFD                                       ; 0905
                                                       11  12 00111         BNEQ     8$                                        ;
                                                0000'  CF  DD 00113         PUSHL    BLDKFDBUF                                 ; 0907
```

```
                                                                              F 15
INSCREATE                                                                     16-Sep-1984 01:49:49    VAX-11 Bliss-32 V4.0-742              Page  27
V04-000          create                                                       14-Sep-1984 12:35:36    [INSTAL.SRC]INSCREATE.B32;1                 (10)

                                         00000000G  00  9F 00117          PUSHAB  INS$G_KFENAM                             :
                          0000V  CF                 02  FB 0011D          CALLS   #2, BUILD_KFD                            :
                                                    05  11 00122          BRB     9$                                       :
                             0C  A8           08    AC  D0 00124 8$:      MOVL    KFD, 12(KFE)                             : 0909
                     08          67                 02  E0 00129 9$:      BBS     #2, (R7), 10$                            : 0914
                     04          67                 0B  E0 0012D          BBS     #11, (R7), 10$                           : 0915
                     3B          67                 03  E1 00131          BBC     #3, (R7), 11$                            : 0916
   0200      8F      00          6E                 00  2C 00135 10$:     MOVC5   #0, (SP), #0, #512, @HDRBLK_BUF          : 0922
                                              0000' DF     0013C
   0200      8F      00          6E                 00  2C 0013F          MOVC5   #0, (SP), #0, #512, @IHDBUF              : 0923
                                              0000' DF     00146
                                                 08 AE  9F 00149          PUSHAB  ALIAS                                    : 0924
                                                 10 AE  9F 0014C          PUSHAB  HDR_VERSION                              :
                                                 1C AE  9F 0014F          PUSHAB  OFFSET                                   :
                                                 24 AE  9F 00152          PUSHAB  VBN                                      :
                                 7E           0000' CF  7D 00155          MOVQ    HDRBLK_BUF, -(SP)                        :
                                         00000000G  00  DD 0015A          PUSHL   INS$GL_KFECHAN                           :
                        00000000G  00               07  FB 00160          CALLS   #7, IMG$DECODE_IHD                       :
                                   5A               50  D0 00167          MOVL    R0, STATUS                               :
                                   03               5A  E8 0016A          BLBS    STATUS, 11$                              : 0926
                                                  044A  31 0016D          BRW     63$                                      :
                     04            67               02  E0 00170 11$:     BBS     #2, (R7), 12$                            : 0933
                     2C            67               0B  E1 00174          BBC     #11, (R7), 14$                           :
                                   50         0000' CF  D0 00178 12$:     MOVL    IHDBUF, R0                               : 0940
                                   51            02 A0  3C 0017D          MOVZWL  2(R0), R1                                :
                5C   AE            50               51  C1 00181          ADDL3   R1, R0, ACTIVOFF                         :
                                   50            5C BE  D0 00186          MOVL    @ACTIVOFF, TFR1                          : 0941
                        00000000G  8F               50  D1 0018A          CMPL    TFR1, #P1SYSVECTORS+360                  : 0942
                                                    09  13 00191          BEQL    13$                                      :
                        80000168   8F               50  D1 00193          CMPL    TFR1, #-2147483288                       : 0944
                                                    08  12 0019A          BNEQ    14$                                      :
                                   50 00000000G  8F     D0 0019C 13$:     MOVL    #INS$_IMGTRACED, R0                      : 0946
                                                    04     001A3          RET                                              :
                     0C            67               03  E0 001A4 14$:     BBS     #3, (R7), 15$                            : 0949
                18   A8 00000000G  00               08  28 001A8          MOVC3   #8, INS$G_KFENAM+36, 24(KFE)             : 0951
                                                  03F2  31 001B1          BRW     62$                                      :
                                   5B            08 AE  3C 001B4 15$:     MOVZWL  ALIAS, R11                               : 0970
                                                    0A  13 001B8          BEQL    16$                                      :
                                   01               5B  B1 001BA          CMPW    R11, #1                                  : 0972
                                                    05  13 001BD          BEQL    16$                                      :
                                   02               5B  B1 001BF          CMPW    R11, #2                                  : 0974
                                                    25  12 001C2          BNEQ    18$                                      :
                                   67            80 8F  88 001C4 16$:     BISB2   #128, (R7)                               : 0982
                     13 00000000G  00               06  E1 001C8          BBC     #6, INS$GL_CTLMSK+1, 17$                 : 0983
                        00000000G  00            40 8F  8A 001D0          BICB2   #64, INS$GL_CTLMSK+1                     : 0986
                                   67               10  8A 001D8          BICB2   #16, (R7)                                : 0987
                        00000000G  00            80 8F  88 001DB          BISB2   #128, INS$GL_CTLMSK+2                    : 0988
                               2A  A8               5B  B0 001E3 17$:     MOVW    R11, 42(KFE)                             : 0990
                                                    4D  11 001E7          BRB     22$                                      : 0969
                                   50         0000' CF  D0 001E9 18$:     MOVL    IHDBUF, R0                               : 0999
                                   51            0E A0  9E 001EE          MOVAB   14(R0), R1                               :
                                                    52  D4 001F2          CLRL    R2                                       : 1008
                                   02            11 A0  91 001F4          CMPB    17(R0), #2                               :
                                                    02  12 001F8          BNEQ    19$                                      :
                                                    52  D6 001FA          INCL    R2                                       :
             67      01            01               52  F0 001FC 19$:     INSV    R2, #1, #1, (R7)                         :
                     31            67               04  E1 00201          BBC     #4, (R7), 22$                            : 1010
```

```
                         3230  8F          0C   A0  B1 00205         CMPW     12(R0), #12848                      ; 1019
                                                0D  12 0020B         BNEQ     20$                                 ;
                               30               61  91 0020D         CMPB     (R1), #48                           ; 1023
                                                08  1A 00210         BGTRU    20$                                 ;
                                                08  12 00212         BNEQ     21$                                 ; 1026
                               35          01   A1  91 00214         CMPB     1(R1), #53                          ; 1028
                                                05  1B 00218         BLEQU    21$                                 ;
                               50          44   8F  9A 0021A  20$:   MOVZBL   #68, R0                             ; 1031
                                                04     0021E         RET                                          ;
                                           28   A0  D5 0021F  21$:   TSTL     40(R0)                              ; 1037
                                                12  13 00222         BEQL     22$                                 ;
                    00000000G  8F          28   A0  D1 00224         CMPL     40(R0), #SYS$K_VERSION              ; 1039
                                                08  13 0022C         BEQL     22$                                 ;
                               50  00000000G    8F  D0 0022E         MOVL     #INS$_SYSVERDIF, R0                 ; 1040
                                                04     00235         RET                                          ;
                 22 00000000G  00               01  E1 00236  22$:   BBC      #1, INS$GL_CTLMSK+2, 24$            ; 1047
       2B        00            6E               00  2C 0023E         MOVC5    #0, (SP), #0, #43, GBLSECNAM        ; 1056
                                           3C   AE     00243                                                      ;
                                           68   AE  D4 00245         CLRL     GBLSECNAM_DSC                       ; 1057
                         6C    AE          3C   AE  9E 00248         MOVAB    GBLSECNAM, GBLSECNAM_DSC+4          ; 1058
                                           68   AE  9F 0024D         PUSHAB   GBLSECNAM_DSC                       ; 1059
                        0000V  CF               01  FB 00250         CALLS    #1, INS$BLD_GBLSECNAM               ;
                                                67  95 00255         TSTB     (R7)                                ; 1061
                                                03  19 00257         BLSS     23$                                 ;
                                              00DA  31 00259         BRW      37$                                 ;
                                                5B  D5 0025C  23$:   TSTL     R11                                 ; 1064
                                                1D  13 0025E         BEQL     26$                                 ;
                 03 00000000G  00               01  E0 00260  24$:   BBS      #1, INS$GL_CTLMSK+2, 25$            ; 1067
                                              013B  31 00268         BRW      44$                                 ;
                    00000000G  00               02  8A 0026B  25$:   BICB2    #2, INS$GL_CTLMSK+2                 ; 1070
                               67               20  8A 00272         BICB2    #32, (R7)                           ; 1071
                               50  00000000G    8F  D0 00275         MOVL     #INS$_NOSHRD, R0                    ; 1073
                                                04     0027C         RET                                          ;
                                           10   AE  9F 0027D  26$:   PUSHAB   IS_SHRMEM                           ; 1087
                                           6C   AE  9F 00280         PUSHAB   GBLSECNAM_DSC                       ;
                        0000V  CF               02  FB 00283         CALLS    #2, CHECK_SHMIDENT                  ;
                               5A               50  D0 00288         MOVL     R0, STATUS                          ;
                               03               5A  E8 0028B         BLBS     STATUS, 27$                         ; 1088
                                              0329  31 0028E         BRW      63$                                 ;
       67        01            06          10   AE  F0 00291  27$:   INSV     IS_SHRMEM, #6, #1, (R7)             ; 1089
                         28    A8               01  90 00297         MOVB     #1, 40(KFE)                         ; 1098
                               50       0000'   CF  D0 0029B         MOVL     IHDBUF, R0                          ; 1100
                                                52  D4 002A0         CLRL     R2                                  ;
                 09      18    A0               0E  E0 002A2         BBS      #14, 24(R0), 28$                    ;
                                                52  D6 002A7         INCL     R2                                  ;
                         2C    A8          1C   A0  D0 002A9         MOVL     28(R0), 44(KFE)                     ; 1102
                                                03  11 002AE         BRB      29$                                 ;
                                           2C   A8  D4 002B0  28$:   CLRL     44(KFE)                             ; 1104
                               04          15   A0  91 002B3  29$:   CMPB     21(R0), #4                          ; 1109
                                                04  13 002B7         BEQL     30$                                 ;
                                                51  D4 002B9         CLRL     N_DSC                               ; 1111
                                                04  11 002BB         BRB      31$                                 ;
                               51          E0   8F  9A 002BD  30$:   MOVZBL   #224, N_DSC                         ; 1113
                               51               50  C0 002C1  31$:   ADDL2    R0, R1                              ; 1123
                               0C               52  E9 002C4         BLBC     R2, 32$                             ;
                               52        00F4   C1  3C 002C7         MOVZWL   244(R1), VBN                        ;
                               51        00F6   C1  3C 002CC         MOVZWL   246(R1), PAGCNT                     ; 1124
```

```
                                          0B  11 002D1         BRB     33$                                  ; 1115
                         52       00EE    C1  3C 002D3 32$:    MOVZWL  238(R1), VBN                         ; 1128
                                          52  D6 002D8         INCL    VBN
                         51         0E    A0  3C 002DA         MOVZWL  14(R0), PAGCNT                       ; 1129
                                          15  12 002DE 33$:    BNEQ    34$                                  ; 1137
               00000000G 00         40    8F  88 002E0         BISB2   #64, INS$GL_CTLMSK+2                 ; 1140
               00000000G 00               02  8A 002E8         BICB2   #2, INS$GL_CTLMSK+2                  ; 1141
                         67         60    8F  8A 002EF         BICB2   #96, (R7)                            ; 1143
                                          3F  11 002F3         BRB     36$                                  ; 1137
                         51               07  C0 002F5 34$:    ADDL2   #7, PAGCNT                           ; 1147
                         51               08  C6 002F8         DIVL2   #8, PAGCNT                           ; 1148
                         59       C001    8F  3C 002FB         MOVZWL  #49153, CRESECFLG                    ; 1149
            03 00000000G 00               02  E1 00300         BBC     #2, INS$GL_CTLMSK+2, 35$             ; 1152
                         59               08  88 00308         BISB2   #8, CRESECFLG                        ; 1154
                                          7E  7C 0030B 35$:    CLRQ    -(SP)                                ; 1169
                                          06  BB 0030D         PUSHR   #^M<R1,R2>
                            00000000G     00  DD 0030F         PUSHL   INS$GL_KFECHAN
                                          7E  D4 00315         CLRL    -(SP)
                                    28    A8  9F 00317         PUSHAB  40(KFE)
                                    98    AD  9F 0031A         PUSHAB  GBLSECNAM_DSC
                                          59  DD 0031D         PUSHL   CRESECFLG
                                          03  DD 0031F         PUSHL   #3
                                          7E  7C 00321         CLRQ    -(SP)
               00000000G 00               0C  FB 00323         CALLS   #12, SYS$CRMPSC
                         5A               50  D0 0032A         MOVL    R0, STATUS
                         3B               5A  E9 0032D         BLBC    STATUS, 40$                          ; 1170
                      12 A8               01  B0 00330         MOVW    #1, 18(KFE)                          ; 1172
                                          70  11 00334 36$:    BRB     44$
                                          59  D4 00336 37$:    CLRL    CRESECFLG                            ; 1184
                      28 A8               01  90 00338         MOVB    #1, 40(KFE)                          ; 1191
                         50       0000'   CF  D0 0033C         MOVL    IHDBUF, R0                           ; 1192
                      2C A8         24    A0  D0 00341         MOVL    36(R0), 44(KFE)
                 13      67               01  E1 00346         BBC     #1, (R7), 39$                        ; 1193
                         07         23    A0  93 0034A         BITB    35(R0), #7                           ; 1196
                                          03  12 0034E         BNEQ    38$
                                    2C    A8  D4 00350         CLRL    44(KFE)                              ; 1198
  51       23     A0     03               00  EF 00353 38$:    EXTZV   #0, #3, 35(R0), R1                   ; 1199
                      28 A8               51  90 00359         MOVB    R1, 40(KFE)
                                    10    AE  9F 0035D 39$:    PUSHAB  IS_SHRMEM                            ; 1206
                                    6C    AE  9F 00360         PUSHAB  GBLSECNAM_DSC
                   0000V CF               02  FB 00363         CALLS   #2, CHECK_SHMIDENT
                         5A               50  D0 00368         MOVL    R0, STATUS
                         03               5A  E8 0036B 40$:    BLBS    STATUS, 41$                          ; 1207
                                     0249 31  0036E            BRW     63$
  67              01     06         10    AE  F0 00371 41$:    INSV    IS_SHRMEM, #6, #1, (R7)              ; 1208
                         2B         10    AE  E9 00377         BLBC    IS_SHRMEM, 44$                       ; 1209
                  27     67               01  E0 0037B         BBS     #1, (R7), 44$
                         50       0000'   CF  D0 0037F         MOVL    IHDBUF, R0                           ; 1217
                         51         08    A0  3C 00384         MOVZWL  8(R0), R1
                                          0D  12 00388         BNEQ    42$
                         51         06    A0  3C 0038A         MOVZWL  6(R0), R1                            ; 1221
                         50               51  C0 0038E         ADDL2   R1, R0
                         50         3A    A0  D0 00391         MOVL    58(R0), R0                           ; 1222
                                          07  11 00395         BRB     43$
                         50               51  C0 00397 42$:    ADDL2   R1, R0                               ; 1227
                         50         26    A0  D0 0039A         MOVL    38(R0), R0                           ; 1228
                      2C A8               50  D0 0039E 43$:    MOVL    R0, 44(KFE)                          ; 1217
```

```
                     28   A8             01  90 003A2       MOVB    #1, 40(KFE)                              : 1231
             2D      67                  04  E1 003A6 44$:  BBC     #4, (R7), 45$                            : 1243
                     24   AE     0200    8F  3C 003AA       MOVZWL  #512, BLDHDR_LEN                         : 1246
                                   2C    AE  9F 003B0       PUSHAB  BLDHDR                                   : 1247
                                   28    AE  9F 003B3       PUSHAB  BLDHDR_LEN                               :
                  00000000G  00          02  FB 003B6       CALLS   #2, LIB$GET_VM                           :
                             7A          50  E9 003BD       BLBC    STATUS, 49$                              :
    24   AE          00      6E          00  2C 003C0       MOVC5   #0, (SP), #0, BLDHDR_LEN, @BLDHDR        : 1249
                                   2C    BE     003C6                                                        :
             2C   BE   0000'     DF   0000'  DF  28 003C8       MOVC3   @IHDBUF, @IHDBUF, @BLDHDR                : 1251
                  30   AE   0000'    DF  3C 003D1       MOVZWL  @IHDBUF, BLDHDR_SIZ                      : 1252
                                   67    95 003D7 45$:  TSTB    (R7)                                     : 1255
                                   03    18 003D9       BGEQ    46$                                      :
                                 01AC    31 003DB       BRW     61$                                      :
  0200   8F          00      6E          00  2C 003DE 46$:  MOVC5   #0, (SP), #0, #512, @ISDBUF              : 1265
                               0000'     DF     003E5                                                        :
                             6E   0000'  CF  D0 003E8       MOVL    ISDBUF, (SP)                             : 1267
                                   0C    AE  DD 003ED 47$:  PUSHL   HDR_VERSION                              :
                                   04    AE  DD 003F0       PUSHL   4(SP)                                    :
                                   1C    AE  9F 003F3       PUSHAB  OFFSET                                   : 1266
                                   24    AE  9F 003F6       PUSHAB  VBN                                      :
                             7E   0000'  CF  7D 003F9       MOVQ    HDRBLK_BUF, -(SP)                        :
                            00000000G    00  DD 003FE       PUSHL   INS$GL_KFECHAN                           :
                  00000000G  00          07  FB 00404       CALLS   #7, IMG$GET_NEXT_ISD                     :
                             5A          50  D0 0040B       MOVL    R0, STATUS                               :
                             03          5A  E8 0040E       BLBS    STATUS, 48$                              :
                                 00F7    31 00411       BRW     59$                                      :
             66              67          04  E1 00414 48$:  BBC     #4, (R7), 53$                            : 1270
                             50   0000'  DF  3C 00418       MOVZWL  @ISDBUF, R0                              : 1276
                             50     30   AE  C0 0041D       ADDL2   BLDHDR_SIZ, R0                           :
                     24      AE          50  D1 00421       CMPL    R0, BLDHDR_LEN                           :
                             40          15 00425       BLEQ    52$                                      :
             20   AE   24    AE          01  78 00427       ASHL    #1, BLDHDR_LEN, NEW_BLDHDR_LEN           : 1283
                                   1C    AE  9F 0042D       PUSHAB  NEW_BLDHDR                               : 1284
                                   24    AE  9F 00430       PUSHAB  NEW_BLDHDR_LEN                           :
                  00000000G  00          02  FB 00433       CALLS   #2, LIB$GET_VM                           :
                             1C          50  E9 0043A 49$:  BLBC    STATUS, 50$                              :
    20   AE          00      6E          00  2C 0043D       MOVC5   #0, (SP), #0, NEW_BLDHDR_LEN, @NEW_BLDHDR : 1285
                                   1C    BE     00443                                                        :
             1C   BE   2C    BE     30   AE  28 00445       MOVC3   BLDHDR_SIZ, @BLDHDR, @NEW_BLDHDR         : 1286
                                   2C    AE  9F 0044C       PUSHAB  BLDHDR                                   : 1287
                                   28    AE  9F 0044F       PUSHAB  BLDHDR_LEN                               :
                  00000000G  00          02  FB 00452       CALLS   #2, LIB$FREE_VM                          :
                             01          50  E8 00459 50$:  BLBS    STATUS, 51$                              :
                                         04 0045C       RET                                              :
                     2C      AE     1C   AE  D0 0045D 51$:  MOVL    NEW_BLDHDR, BLDHDR                       : 1288
                     24      AE     20   AE  D0 00462       MOVL    NEW_BLDHDR_LEN, BLDHDR_LEN               : 1289
             50      2C      AE     30   AE  C1 00467 52$:  ADDL3   BLDHDR_SIZ, BLDHDR, R0                   : 1292
             60   0000'      DF   0000'  DF  28 0046D       MOVC3   @ISDBUF, @ISDBUF, (R0)                   :
                             50   0000'  DF  3C 00475       MOVZWL  @ISDBUF, R0                              : 1293
                     30      AE          50  C0 0047A       ADDL2   R0, BLDHDR_SIZ                           :
             74 00000000G    00          01  E1 0047E 53$:  BBC     #1, INS$GL_CTLMSK+2, 58$                 : 1299
                             50   0000'  CF  D0 00486       MOVL    ISDBUF, R0                               : 1303
                             6B     08   A0  E8 0048B       BLBS    8(R0), 58$                               : 1305
             66      08      A0          02  E0 0048F       BBS     #2, 8(R0), 58$                           :
             61      08      A0          01  E0 00494       BBS     #1, 8(R0), 58$                           : 1306
             59      08      A0 FFFFFFF7 8F  CB 00499       BICL3   #-9, 8(R0), CRESECFLG                    : 1312
```

```
                                       59     C001   8F  A8 004A2        BISW2   #49153, CRESECFLG                       : 1314
                         08        0A  A0            02  E0 004A7        BBS     #2, 10(R0), 54$                         : 1316
                                       0C            67  E9 004AC        BLBC    (R7), 55$                               : 1317
                         07        08  A0            03  E0 004AF        BBS     #3, 8(R0), 55$                          :
                                       59 00040040   8F  C8 004B4 54$:   BISL2   #262208, CRESECFLG                      : 1321
                                       7E       07   A0  9A 004BB 55$:   MOVZBL  7(R0), -(SP)                            : 1338
                                                     7E  D4 004BF        CLRL    -(SP)                                   :
                                                0C   A0  DD 004C1        PUSHL   12(R0)                                  :
                                       7E       02   A0  3C 004C4        MOVZWL  2(R0), -(SP)                            :
                                         00000000G   00  DD 004C8        PUSHL   INS$GL_KFECHAN                          :
                                                     7E  D4 004CE        CLRL    -(SP)                                   :
                                                28   A8  9F 004D0        PUSHAB  40(KFE)                                 :
                                                98   AD  9F 004D3        PUSHAB  GBLSECNAM_DSC                           :
                                                     59  DD 004D6        PUSHL   CRESECFLG                               :
                                                     03  DD 004D8        PUSHL   #3                                      :
                                                5C   AE  9F 004DA        PUSHAB  RETADR                                  :
                                                     7E  D4 004DD        CLRL    -(SP)                                   :
                            00000000G  00            0C  FB 004DF        CALLS   #12, SYS$CRMPSC                         :
                                       5A            50  D0 004E6        MOVL    R0, STATUS                              :
                                       03            5A  E8 004E9        BLBS    STATUS, 57$                             : 1339
                                                   00CB  31 004EC 56$:   BRW     63$                                     :
                                                68   AE  9F 004EF 57$:   PUSHAB  GBLSECNAM_DSC                           : 1342
                                0000V  CF            01  FB 004F2        CALLS   #1, INS$BLD_GBLSECNAM                   :
                                                12   A8  B6 004F7        INCW    18(KFE)                                 : 1343
                                       6E    0000'   CF  D0 004FA 58$:   MOVL    ISDBUF, (SP)                            : 1350
    0200   8F            00            6E            00  2C 004FF        MOVC5   #0, (SP), #0, #512, @0(SP)              :
                                                00   BE     00506                                                        :
                                                   FEE2  31 00508        BRW     47$                                     : 1266
                             084D8640  8F            5A  D1 0050B 59$:   CMPL    STATUS, #139298368                      : 1353
                                                     D8  12 00512        BNEQ    56$                                     :
                         18 00000000G  00            01  E1 00514        BBC     #1, INS$GL_CTLMSK+2, 60$                : 1359
                                                12   A8  B5 0051C        TSTW    18(KFE)                                 :
                                                     13  12 0051F        BNEQ    60$                                     :
                            00000000G  00       40   8F  88 00521        BISB2   #64, INS$GL_CTLMSK+2                    : 1362
                            00000000G  00            02  8A 00529        BICB2   #2, INS$GL_CTLMSK+2                     : 1363
                                       67       60   8F  8A 00530        BICB2   #96, (R7)                               : 1365
                         52            67            04  E1 00534 60$:   BBC     #4, (R7), 61$                           : 1368
                         56        30  AE            10  C1 00538        ADDL3   #16, BLDHDR_SIZ, LENGTH                 : 1377
                                                28   AE  9F 0053D        PUSHAB  KFRH                                    : 1378
                                                     56  DD 00540        PUSHL   LENGTH                                  :
                                0000V  CF            02  FB 00542        CALLS   #2, ALLOC_PAGED                         :
                                       73            50  E9 00547        BLBC    STATUS, 64$                             :
                                       57       28   AE  D0 0054A        MOVL    KFRH, R7                                : 1379
           56            00            6E            00  2C 0054E        MOVC5   #0, (SP), #0, LENGTH, (R7)              :
                                                     67     00553                                                        :
                                   04  A7            5B  B0 00554        MOVW    R11, 4(R7)                              : 1381
                                   08  A7            56  B0 00558        MOVW    LENGTH, 8(R7)                           : 1382
                                   0A  A7            26  90 0055C        MOVB    #38, 10(R7)                             : 1383
                                   0B  A7       0C   AE  90 00560        MOVB    HDR_VERSION, 11(R7)                     : 1384
                                   1C  A8       0C   A7  9E 00565        MOVAB   12(R7), 28(KFE)                         : 1385
                    0C   A7        2C  BE       30   AE  28 0056A        MOVC3   BLDHDR_SIZ, @BLDHDR, 12(R7)             : 1386
                         50            57       30   AE  C1 00571        ADDL3   BLDHDR_SIZ, R7, R0                      : 1387
                                       67       0C   A0  9E 00576        MOVAB   12(R0), (R7)                            :
                                                2C   AE  9F 0057A        PUSHAB  BLDHDR                                  : 1388
                                                34   AE  9F 0057D        PUSHAB  BLDHDR_SIZ                              :
                            00000000G  00            02  FB 00580        CALLS   #2, LIB$FREE_VM                         :
                                       33            50  E9 00587        BLBC    STATUS, 64$                             :
```

```
                                                                     K 15
INSCREATE                                                            16-Sep-1984 01:49:49    VAX-11 Bliss-32 V4.0-742             Page  32
V04-000          create                                              14-Sep-1984 12:35:36    [INSTAL.SRC]INSCREATE.B32;1               (10)

                                 34   A8           01  B0 0058A 61$:    MOVW     #1, 52(KFE)                                    ; 1392
                      50         04   AE           04  C1 0058E         ADDL3    #4, CCB, R0                                    ; 1393
                                      52           60  D0 00593         MOVL     (R0), WCB                                      ;
                                 18   A8           52  D0 00596         MOVL     WCB, 24(KFE)                                   ; 1394
                                      50           52  D0 0059A         MOVL     WCB, R0                                        ; 1398
                                         00000000G 00  16 0059D         JSB      MMG$RET_BYT_QUOTA                              ;
                                                0E A2  B6 005A3         INCW     14(WCB)                                        ; 1399
                                                0C AC  DD 005A6 62$:    PUSHL    KFD_INSERT_ADR                                 ; 1403
                                             0000' CF  DD 005A9         PUSHL    BLDKFDBUF                                      ;
                                                04 AC  DD 005AD         PUSHL    HASH_INDEX                                     ;
                                                   58  DD 005B0         PUSHL    KFE                                            ;
                               0000V  CF           04  FB 005B2         CALLS    #4, ENTER_KFE                                  ;
                                      5A           50  D0 005B7         MOVL     R0, STATUS                                     ;
                                      50           5A  D0 005BA 63$:    MOVL     STATUS, R0                                     ; 1405
                                                   04     005BD 64$:    RET                                                     ; 1406

; Routine Size:  1470 bytes,    Routine Base:  $CODE$ + 0105


; 1041          1407  1
; 1043          1408  1 %SBTTL 'alloc_paged  Allocate memory from paged pool';
; 1044          1409  1
; 1045          1410  1 ROUTINE  ALLOC_PAGED (LEN, ADR) =
; 1046          1411  2 BEGIN
; 1047          1412  2 !+++
; 1048          1413  2 !
; 1049          1414  2 !  FUNCTIONAL DESCRIPTION:
; 1050          1415  2 !
; 1051          1416  2 !       Jacket routine for calling paged pool allocation routine.
; 1052          1417  2 !       Specify the length of block required and get the address of
; 1053          1418  2 !       allocated block returned in ADR.
; 1054          1419  2 !
; 1055          1420  2 !---
; 1056          1421  2
; 1057          1422  2 GLOBAL REGISTER
; 1058          1423  2     LENGTH = 1,         ! Length to allocate
; 1059          1424  2     ENTRY_BLOCK = 2;    ! Address of allocated block
; 1060          1425  2
; 1061          1426  2 LOCAL
; 1062          1427  2     STATUS;
; 1063          1428  2
; 1064          1429  2 LENGTH = .LEN;          ! Place length into R1
; 1065          1430  2
; 1066          1431  2 STATUS = EXE$ALOPAGED ();       ! Allocate from paged pool
; 1067          1432  2
; 1068          1433  2 .ADR = .ENTRY_BLOCK;    ! Return address of block
; 1069          1434  2
; 1070          1435  2 IF NOT .STATUS
; 1071          1436  2     THEN STATUS = INS$_NOPAGEDYN;
; 1072          1437  2
; 1073          1438  2 RETURN .STATUS;
; 1074          1439  1 END;                    ! Routine ALLO_PAGED



                                            0FFC 00000 ALLOC_PAGED:
                                                             .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11                ; 1410
                                51        04  AC  D0 00002         MOVL    LEN, LENGTH                                         ; 1429
                                  00000000G  00  16 00006         JSB     EXE$ALOPAGED                                        ; 1431
                          08  BC              52  D0 0000C         MOVL    ENTRY_BLOCK, @ADR                                   ; 1433
                              07              50  E8 00010         BLBS    STATUS, 1$                                          ; 1435
                              50  00000000G  8F  D0 00013         MOVL    #INS$_NOPAGEDYN, STATUS                             ; 1436
                                                  04 0001A 1$:     RET                                                         ; 1439

; Routine Size:  27 bytes,    Routine Base:  $CODE$ + 06C3


; 1075          1440  1
```

```
1 %SBTTL 'find_kfd  Locate Device, Directory, Type block for KFE';
1
1 ROUTINE  FIND_KFD (NAMBLK, INSERT_KFD_ADR) =
2 BEGIN
2 !+++
2 !
2 !  FUNCTIONAL DESCRIPTION:
2 !
2 !       Given a name block for a file, figure out which KFD list it
2 !       would be in.  If it is in a KFD list, return the address
2 !       of the KFD in R0.  If the KFD doesn't exist, then return 0
2 !       and place the address of where the KFD should go when it's
2 !       created into INSERT_KFD_ADR.
2 !
2 !---
2 MAP
2     NAMBLK : REF BBLOCK;
2
2 BIND
2     INSERT_KFD = .INSERT_KFD_ADR,
2     KFPB = EXE$GL_KNOWN_FILES : REF BBLOCK;
2
2 LOCAL
2     KFD : REF BBLOCK,
2     DDTSTR : BBLOCK [NAM$C_MAXRSS],
2     DDT_DSC : $BBLOCK [DSC$C_S_BLN],
2     PRV_KFD;                          ! Previous KFD
2
2 IF .KFPB EQL 0                        ! There is no pointer block yet
2 THEN
3     BEGIN
3     INSERT_KFD = 0;
3     RETURN 0;
3     END;
2
2 IF .KFPB [KFPB$L_KFDLST] EQL 0  ! If there are no KFDs in list
2 THEN
3     BEGIN                             ! Make it the first
3     INSERT_KFD = KFPB [KFPB$L_KFDLST];
3     RETURN 0;                         ! There are no KFDs
2     END;
2
2 !
2 !   Build an ASCII string of the concatenated Device, Directory
2 !   Type strings.
2 !
2 DDT_DSC [DSC$W_LENGTH] = .NAMBLK [NAM$B_DEV] + .NAMBLK [NAM$B_DIR] +
2                          .NAMBLK [NAM$B_TYPE];   ! Length of DDT string
2
2 DDT_DSC [DSC$A_POINTER] = DDTSTR;
2 DDT_DSC [DSC$A_POINTER] = CH$MOVE (.NAMBLK [NAM$B_DEV], .NAMBLK [NAM$L_DEV],
2                                   .DDT_DSC [DSC$A_POINTER]);
2 DDT_DSC [DSC$A_POINTER] = CH$MOVE (.NAMBLK [NAM$B_DIR], .NAMBLK [NAM$L_DIR],
2                                   .DDT_DSC [DSC$A_POINTER]);
2 DDT_DSC [DSC$A_POINTER] = CH$MOVE (.NAMBLK [NAM$B_TYPE], .NAMBLK [NAM$L_TYPE],
2                                   .DDT_DSC [DSC$A_POINTER]);
2
```

```
; 1134          1498  2 DDT_DSC [DSC$A_POINTER] = DDTSTR;
; 1135          1499  2 INS$CVT_DIR (DDT_DSC);           ! Convert and compress directory brackets
; 1136          1500  2 !
; 1137          1501  2 !   Traverse the KFD list to find a KFD block with a matching DDT string.
; 1138          1502  2 !   If no match is found, record address of block after which a new KFD
; 1139          1503  2 !   block containing the new DDT string should be inserted.
; 1140          1504  2 !
; 1141          1505  2 PRV_KFD = KFPB [KFPB$L_KFDLST];
; 1142          1506  2 KFD = .KFPB [KFPB$L_KFDLST];
; 1143          1507  2 WHILE .KFD NEQ 0 DO              ! Single linked list ending in zero
; 1144          1508  3     BEGIN
; 1145          1509  3     CASE CH$COMPARE (.DDT_DSC [DSC$W_LENGTH], DDTSTR,
; 1146          1510  3             .KFD [KFD$B_DDTSTRLEN], KFD [KFD$T_DDTSTR], %C' ')
; 1147          1511  3     FROM -1 TO 1 OF               ! Either less than, equal to, or greater than
; 1148          1512  3         SET
; 1149          1513  3
; 1150          1514  3         [-1]:   ! Less than, therefore its not in the list
; 1151          1515  4             BEGIN
; 1152          1516  4             INSERT_KFD = .PRV_KFD;        ! Return Previous KFD to caller
; 1153          1517  4             RETURN 0;                     ! Return KFD not found
; 1154          1518  3             END;
; 1155          1519  3
; 1156          1520  3         [0] :
; 1157          1521  4             BEGIN
; 1158          1522  4             INSERT_KFD = 0;               ! Return a ZERO to caller
; 1159          1523  4             RETURN .KFD;                  ! Return KFD found
; 1160          1524  3             END;
; 1161          1525  3
; 1162          1526  3         [1] :   ! Greater than,
; 1163          1527  4             BEGIN
; 1164          1528  4             PRV_KFD = .KFD;               ! Current KFD now becomes previous
; 1165          1529  4             KFD = .KFD [KFD$L_LINK];      ! Follow link for next current KFD
; 1166          1530  3             END;
; 1167          1531  3         TES;
; 1168          1532  2     END;                          ! WHILE traversing KFD list
; 1169          1533  2
; 1170          1534  2 !
; 1171          1535  2 !   Traversed whole list without finding match or finding where it
; 1172          1536  2 !   should fit in list, so put it at the end
; 1173          1537  2 !
; 1174          1538  2 INSERT_KFD = .PRV_KFD;
; 1175          1539  2 RETURN 0;
; 1176          1540  1 END;                              ! Routine find_kfd



                                        01FC 00000 FIND_KFD:
                                                          .WORD   Save R2,R3,R4,R5,R6,R7,R8                        ; 1443
                    58 0000000G    00   9E 00002          MOVAB   KFPB, R8                                         ;
                    5E      FEF8   CE   9E 00009          MOVAB   -264(SP), SP                                     ;
                    57        08   AC   D0 0000E          MOVL    INSERT_KFD_ADR, R7                               ; 1460
                    50             68   D0 00012          MOVL    KFPB, R0                                         ; 1469
                                   04   12 00015          BNEQ    1$                                               ;
                                   67   D4 00017          CLRL    (R7)                                             ; 1472
                                   07   11 00019          BRB     2$                                               ; 1473
```

```
                                          60  D5 0001B 1$:      TSTL    (R0)                                   ; 1476
                                          06  12 0001D          BNEQ    3$                                     ;
                              67          50  D0 0001F          MOVL    R0, (R7)                               ; 1479
                                        0080  31 00022 2$:      BRW     7$                                     ; 1480
                              56    04    AC  D0 00025 3$:      MOVL    NAMBLK, R6                             ; 1487
                              50    39    A6  9A 00029          MOVZBL  57(R6), R0                             ;
                              51    3A    A6  9A 0002D          MOVZBL  58(R6), R1                             ;
                              50          51  C0 00031          ADDL2   R1, R0                                 ;
                              52    3C    A6  9A 00034          MOVZBL  60(R6), R2                             ; 1488
                 6E           50          52  A1 00038          ADDW3   R2, R0, DDT_DSC                        ;
                        04    AE    08    AE  9E 0003C          MOVAB   DDTSTR, DDT_DSC+4                      ; 1490
                              50    39    A6  9A 00041          MOVZBL  57(R6), R0                             ; 1491
            04   BE     44    B6          50  28 00045          MOVC3   R0, @68(R6), @DDT_DSC+4                ; 1492
                        04    AE          53  D0 0004B          MOVL    R3, DDT_DSC+4                          ;
                              50    3A    A6  9A 0004F          MOVZBL  58(R6), R0                             ; 1493
            04   BE     48    B6          50  28 00053          MOVC3   R0, @72(R6), @DDT_DSC+4                ; 1494
                        04    AE          53  D0 00059          MOVL    R3, DDT_DSC+4                          ;
                              50    3C    A6  9A 0005D          MOVZBL  60(R6), R0                             ; 1495
            04   BE     50    B6          50  28 00061          MOVC3   R0, @80(R6), @DDT_DSC+4                ; 1496
                        04    AE          53  D0 00067          MOVL    R3, DDT_DSC+4                          ;
                        04    AE    08    AE  9E 0006B          MOVAB   DDTSTR, DDT_DSC+4                      ; 1498
                                          5E  DD 00070          PUSHL   SP                                     ; 1499
                   00000000G  00          01  FB 00072          CALLS   #1, INS$CVT_DIR                        ;
                              50          68  D0 00079          MOVL    KFPB, R0                               ; 1505
                              56          50  D0 0007C          MOVL    R0, PRV_KFD                            ;
                              54          60  D0 0007F          MOVL    (R0), KFD                              ; 1506
                                          1E  13 00082 4$:      BEQL    6$                                     ; 1507
                              50    10    A4  9A 00084          MOVZBL  16(KFD), R0                            ; 1510
       50        20     08    AE          6E  2D 00088          CMPC5   DDT_DSC, DDTSTR, #32, R0, 17(KFD)      ;
                                    11    A4     0008E                                                         ;
                                          08  1A 00090          BGTRU   5$                                     ;
                                          0E  1F 00092          BLSSU   6$                                     ;
                                          67  D4 00094          CLRL    (R7)                                   ; 1522
                              50          54  D0 00096          MOVL    KFD, R0                                ; 1523
                                              04 00099          RET                                            ;
                              56          54  D0 0009A 5$:      MOVL    KFD, PRV_KFD                           ; 1528
                              54          64  D0 0009D          MOVL    (KFD), KFD                             ; 1529
                                          E0  11 000A0          BRB     4$                                     ; 1507
                              67          56  D0 000A2 6$:      MOVL    PRV_KFD, (R7)                          ; 1538
                                          50  D4 000A5 7$:      CLRL    R0                                     ; 1540
                                              04 000A7          RET                                            ;

; Routine Size:  168 bytes,    Routine Base:  $CODE$ + 06DE


; 1177          1541  1
```

```
1 %SBTTL 'build_kfd  Build a Device, Directory, Type block for the KFE';
1
1 ROUTINE  BUILD_KFD (NAMBLK,KFDBUF) : NOVALUE =
2 BEGIN
2 !+++
2 !
2 !  FUNCTIONAL DESCRIPTION:
2 !
2 !       Given the file info in the NAM block, construct a KFD entry.
2 !       A KFD entry is a list head for all known file entries which
2 !       share the same Device, directory and file type.
2 !
2 ! INPUTS:
2 !
2 !       NAMBLK = Address of the NAM block
2 !       KFDBUF = Address of the buffer to build the kfd in
2 !                   (must be KFD$C_LENGTH+NAM$C_MAXRSS in length)
2 !---
2 MAP
2     NAMBLK : REF BBLOCK,
2     KFDBUF : REF $BBLOCK;
2
2 LOCAL
2     DDT_DSC : $BBLOCK [DSC$C_S_BLN],
2     PTR,
2     PTR2,
2     LENGTH;
2
2 DDT_DSC [DSC$W_LENGTH] = .NAMBLK [NAM$B_DEV] + .NAMBLK [NAM$B_DIR] +
2                          .NAMBLK [NAM$B_TYPE];   ! Length of DDT string
2 LENGTH = KFD$C_LENGTH + .DDT_DSC [DSC$W_LENGTH];
2
2 CH$FILL (0, .LENGTH, .KFDBUF);                       ! zero the KFD
2 KFDBUF [KFD$W_SIZE] = .LENGTH;
2 KFDBUF [KFD$B_TYPE] = DYN$C_KFD;
2 KFDBUF [KFD$B_DDTSTRLEN] = .DDT_DSC [DSC$W_LENGTH];
2
2 !
2 !   Build a counted ASCII string of the concatenated Device, Directory
2 !   Type strings.
2 !
2 DDT_DSC [DSC$A_POINTER] = KFDBUF [KFD$T_DDTSTR];
2 KFDBUF [KFD$B_DEVLEN] = .NAMBLK [NAM$B_DEV];
2 DDT_DSC [DSC$A_POINTER] = CH$MOVE (.NAMBLK [NAM$B_DEV], .NAMBLK [NAM$L_DEV],
2                                 .DDT_DSC [DSC$A_POINTER]);
2 KFDBUF [KFD$B_DIRLEN] = .NAMBLK [NAM$B_DIR];
2 DDT_DSC [DSC$A_POINTER] = CH$MOVE (.NAMBLK [NAM$B_DIR], .NAMBLK [NAM$L_DIR],
2                                 .DDT_DSC [DSC$A_POINTER]);
2 DDT_DSC [DSC$A_POINTER] = CH$MOVE (.NAMBLK [NAM$B_TYPE], .NAMBLK [NAM$L_TYPE],
2                                 .DDT_DSC [DSC$A_POINTER]);
2
2 LENGTH = .DDT_DSC [DSC$W_LENGTH];        ! Save current DDT length
2 DDT_DSC [DSC$A_POINTER] = KFDBUF [KFD$T_DDTSTR];
2 INS$CVT_DIR (DDT_DSC);                   ! Convert and compress directory brackets
2 !
2 ! Calculate amount of string compression that occurred and
2 ! correct the fields in the KFD where appropriate.
```

```
; 1236         1599  2 !
; 1237         1600  2 LENGTH = .LENGTH - .DDT_DSC [DSC$W_LENGTH];
; 1238         1601  2 KFDBUF [KFD$B_DIRLEN] = .KFDBUF [KFD$B_DIRLEN] - .LENGTH;
; 1239         1602  2 KFDBUF [KFD$W_SIZE] = .KFDBUF [KFD$W_SIZE] - .LENGTH;
; 1240         1603  2 KFDBUF [KFD$B_DDTSTRLEN] = .KFDBUF [KFD$B_DDTSTRLEN] - .LENGTH;
; 1241         1604  2 RETURN;
; 1242         1605  1 END;                           ! Routine build_kfd



                                                       03FC 00000 BUILD_KFD:
                                                                         .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9                ; 1544
                                        5E          08   C2 00002        SUBL2    #8, SP                                      ;
                                        57    04    AC   D0 00005        MOVL     NAMBLK, R7                                  ; 1570
                                        50    39    A7   9A 00009        MOVZBL   57(R7), R0                                  ;
                                        51    3A    A7   9A 0000D        MOVZBL   58(R7), R1                                  ;
                                        50          51   C0 00011        ADDL2    R1, R0                                      ;
                                        52    3C    A7   9A 00014        MOVZBL   60(R7), R2                                  ; 1571
                      6E                50          52   A1 00018        ADDW3    R2, R0, DDT_DSC                             ;
                                        58          6E   3C 0001C        MOVZWL   DDT_DSC, LENGTH                             ; 1572
                                        58          11   C0 0001F        ADDL2    #17, LENGTH                                 ;
                                        56    08    AC   D0 00022        MOVL     KFDBUF, R6                                  ; 1574
        58            00                6E          00   2C 00026        MOVC5    #0, (SP), #0, LENGTH, (R6)                  ;
                                                    66      0002B                                                             ;
                                  08    A6          58   B0 0002C        MOVW     LENGTH, 8(R6)                               ; 1575
                                  0A    A6    43    8F   90 00030        MOVB     #67, 10(R6)                                 ; 1576
                                  10    A6          6E   90 00035        MOVB     DDT_DSC, 16(R6)                             ; 1577
                                        59    11    A6   9E 00039        MOVAB    17(R6), R9                                  ; 1583
                                  04    AE          59   D0 0003D        MOVL     R9, DDT_DSC+4                               ;
                                  0E    A6    39    A7   90 00041        MOVB     57(R7), 14(R6)                              ; 1584
                                        50    39    A7   9A 00046        MOVZBL   57(R7), R0                                  ; 1585
                04    BE          44    B7          50   28 0004A        MOVC3    R0, @68(R7), @DDT_DSC+4                     ; 1586
                                  04    AE          53   D0 00050        MOVL     R3, DDT_DSC+4                               ;
                                  0F    A6    3A    A7   90 00054        MOVB     58(R7), 15(R6)                              ; 1587
                                        50    3A    A7   9A 00059        MOVZBL   58(R7), R0                                  ; 1588
                04    BE          48    B7          50   28 0005D        MOVC3    R0, @72(R7), @DDT_DSC+4                     ; 1589
                                  04    AE          53   D0 00063        MOVL     R3, DDT_DSC+4                               ;
                                        50    3C    A7   9A 00067        MOVZBL   60(R7), R0                                  ; 1590
                04    BE          50    B7          50   28 0006B        MOVC3    R0, @80(R7), @DDT_DSC+4                     ; 1591
                                  04    AE          53   D0 00071        MOVL     R3, DDT_DSC+4                               ;
                                        58          6E   3C 00075        MOVZWL   DDT_DSC, LENGTH                             ; 1593
                                  04    AE          59   D0 00078        MOVL     R9, DDT_DSC+4                               ; 1594
                                                    5E   DD 0007C        PUSHL    SP                                          ; 1595
                          000000000G    00          01   FB 0007E        CALLS    #1, INS$CVT_DIR                             ;
                                        50          6E   3C 00085        MOVZWL   DDT_DSC, R0                                 ; 1600
                                        58          50   C2 00088        SUBL2    R0, LENGTH                                  ;
                                  0F    A6          58   82 0008B        SUBB2    LENGTH, 15(R6)                              ; 1601
                                  08    A6          58   A2 0008F        SUBW2    LENGTH, 8(R6)                               ; 1602
                                  10    A6          58   82 00093        SUBB2    LENGTH, 16(R6)                              ; 1603
                                                         04 00097        RET                                                  ; 1605

; Routine Size:  152 bytes,    Routine Base:  $CODE$ + 0786


; 1243         1606  1
```

```
1 %SBTTL 'Enter_kfe  Enter the KFE into the hash table and KFE list';
1
1 ROUTINE  ENTER_KFE (KFE_TMP, HSHIDX, NEWKFD, NEWKFD_INSERT_ADR) =
2 BEGIN
2 !+++
2 !
2 !   FUNCTIONAL DESCRIPTION:
2 !
2 !         Place the KFE into the KFD list and the Hash table list.
2 !         The Hash list is the one used by RMS open to determine if
2 !         the file is installed.  The KFD list is the ordered list
2 !         which is traversed when the known file data base is LISTed.
2 !
2 !         KFE_TMP         Address of temporary block containing copy of KFE
2 !         HSHIDX          Index into hash table where entry should be inserted
2 !         NEWKFD          Address of KFD entry if this KFE was first in a new
2 !                          KFD list
2 !         NEW_KFD_INSERT_ADR
2 !                         Address in KFD list in which to place the new KFD if
2 !                         one was required.
2 !
2 !---
2 MAP
2     KFE_TMP : REF BBLOCK,
2     NEWKFD : REF $BBLOCK,
2     NEWKFD_INSERT_ADR : REF BBLOCK;
2
2 LOCAL
2     HSHTAB : REF VECTOR [,LONG],
2     KFD : REF BBLOCK,
2     KFE : REF $BBLOCK;
2
2 BIND
2     KFPB = EXE$GL_KNOWN_FILES : REF BBLOCK;
2
2 INS$CNVRT_KF_LOCK (LCK$K_EXMODE);       ! Convert protected read to exclusive
2                                         !  to lock out any image activations
2
2 SET_IPL (IPL$_ASTDEL);
2 EXECUTE(ALLOC_PAGED ( .KFE_TMP [KFE$W_SIZE], KFE));
2 CH$MOVE ( .KFE_TMP [KFE$W_SIZE], .KFE_TMP, .KFE);       ! Copy temp to paged pool
2
2 IF .KFPB EQL 0
2 THEN
3     BEGIN
3     !
3     !   Allocate Known file pointer block
3     !
3     EXECUTE(ALLOC_PAGED (KFPB$C_LENGTH, KFPB));
3     CH$FILL (0, KFPB$C_LENGTH, .KFPB);
3     KFPB [KFPB$W_SIZE] = KFPB$C_LENGTH;
3     KFPB [KFPB$B_TYPE] = DYN$C_KFPB;
3
3     !
3     !   NEWKFD_INSERT_ADR must have been zero since there was no header
3     !   block before now.  So the KFD for the KFE being inserted will be
3     !   the first in the list.
```

```
3           !
3           NEWKFD_INSERT_ADR = KFPB [KFPB$L_KFDLST];
3
3           !
3           !   Allocate Hash table
3           !
3           EXECUTE(ALLOC_PAGED (4 * .SGN_B_KFHSHSIZ, KFPB [KFPB$L_KFEHSHTAB]));
3           KFPB [KFPB$W_HSHTABLEN] = .SGN_B_KFHSHSIZ;
3           CH$FILL (0, 4 * .SGN_B_KFHSHSIZ, .KFPB [KFPB$L_KFEHSHTAB]);
2           END;
2
2       HSHTAB = .KFPB [KFPB$L_KFEHSHTAB];
2
2           !
2           !   Search the hash bucket linked list for insertion point
2           !
3           BEGIN
3           LOCAL
3               CMPKFE : REF BBLOCK,
3               PRVKFE : REF BBLOCK;
3
3           PRVKFE = HSHTAB [.HSHIDX];              ! Previous KFE
3           CMPKFE = .HSHTAB [.HSHIDX];             ! Comparison KFE
3           WHILE .CMPKFE NEQ 0 DO                  ! Single linked list ending in zero
4               BEGIN
4               CASE CH$COMPARE (.KFE [KFE$B_FILNAMLEN], KFE [KFE$T_FILNAM],
4                       .CMPKFE [KFE$B_FILNAMLEN], CMPKFE [KFE$T_FILNAM], %C' ')
4               FROM -1 TO 1 OF                     ! Either less than, equal to, or greater than
4                   SET
4
4                   [-1]:       ! Less than, therefore its not in the list, insert here
5                       BEGIN
5                       KFE [KFE$L_HSHLNK] = .PRVKFE [KFE$L_HSHLNK];
5                       PRVKFE [KFE$L_HSHLNK] =  KFE [KFE$L_HSHLNK];
5                       PRVKFE = 0;     ! Mark as inserted
5                       CMPKFE = 0;     ! Terminate traversal
4                       END;
4
4                   [0] :       ! Same file name, place newest in front
5                       BEGIN
5                       KFE [KFE$L_HSHLNK] = .PRVKFE [KFE$L_HSHLNK];
5                       PRVKFE [KFE$L_HSHLNK] =  KFE [KFE$L_HSHLNK];
5                       PRVKFE = 0;     ! Mark as inserted
5                       CMPKFE = 0;     ! Terminate traversal
4                       END;
4
4                   [1] :       ! Greater than,
5                       BEGIN
5                       PRVKFE = .CMPKFE;
5                       CMPKFE = .CMPKFE [KFE$L_HSHLNK];
4                       END;
4                   TES;
3               END;                        ! WHILE traversing hash bucket list
3
3           !
3           !   Have traversed whole list.  If PRVKFE has been set to 0, then
3           !   it was inserted, else it goes at the end.
```

```
3           !
3           IF .PRVKFE NEQ 0
3           THEN
3               PRVKFE [KFE$L_HSHLNK] = .KFE;
2           END;                            ! Block for inserting KFE into Hash bucket list
2
2       KFPB [KFPB$W_KFDLSTCNT] = .KFPB [KFPB$W_KFDLSTCNT] + 1;
2
2       KFD = .KFE [KFE$L_KFD];
2       IF .KFD EQL 0
2       THEN
3           BEGIN
3           EXECUTE(ALLOC_PAGED(.NEWKFD[KFD$W_SIZE],KFD));
3           CH$MOVE(.NEWKFD[KFD$W_SIZE],.NEWKFD,.KFD);              !Copy the KFD
3           KFE [KFE$L_KFD] = .KFD;
3           !
3           !   New KFD must be inserted into list
3           !
3           KFD [KFD$L_LINK] = .NEWKFD_INSERT_ADR [KFD$L_LINK];
3           .NEWKFD_INSERT_ADR = .KFD;
3
3           KFPB [KFPB$W_KFDLSTCNT] = .KFPB [KFPB$W_KFDLSTCNT] + 1;
2           END;
2
2       KFD [KFD$W_REFCNT] = .KFD [KFD$W_REFCNT] + 1;
2
2       !
2       !   Now thread the filename ordered list from the KFD
2       !
2       IF .KFD [KFD$L_KFELIST] EQL 0
2       THEN
2           !
2           !   The list is empty, so make this the first entry
2           !
2           KFD [KFD$L_KFELIST] = .KFE
2       ELSE
2           !
2           !   Must be inserted somewhere in the ordered list of KFEs
2           !
3           BEGIN
3           LOCAL
3               CMPKFE : REF BBLOCK,
3               PRVKFE : REF BBLOCK;
3
3           PRVKFE = .KFD;                          ! Initialize Previous KFE
3                                                   ! *** CAUTION ***
3                                                   ! This assumes kfd$l_kfelist = kfe$l_kfelist
3
3           CMPKFE = .KFD [KFD$L_KFELIST];          ! Comparison KFE
3           WHILE .CMPKFE NEQ 0 DO                  ! Single linked list ending in zero
4               BEGIN
4               CASE CH$COMPARE (.KFE [KFE$B_FILNAMLEN], KFE [KFE$T_FILNAM],
4                       .CMPKFE [KFE$B_FILNAMLEN], CMPKFE [KFE$T_FILNAM], %C' ')
4               FROM -1 TO 1 OF                     ! Either less than, equal to, or greater than
4                   SET
4
```

```
; 1416           1778  4                [-1]:        ! Less than, therefore its not in the list, insert here
; 1417           1779  5                    BEGIN
; 1418           1780  5                    KFE [KFE$L_KFELINK] = .CMPKFE;
; 1419           1781  5                    PRVKFE [KFE$L_KFELINK] = .KFE;
; 1420           1782  5                    PRVKFE = 0;      ! Mark as inserted
; 1421           1783  5                    CMPKFE = 0;      ! Terminate traversal
; 1422           1784  4                    END;
; 1423           1785  4
; 1424           1786  4                [0] :        ! Same file name in same KFD, is a serious bug
; 1425           1787  5                    BEGIN
; 1426           1788  5                    INS$L_INTRNLERR = DUPINKFD_ERR_DSC;
; 1427           1789  5                    INS$CNVRT_KF_LOCK (LCK$K_PRMODE);          ! Convert exclusive to protected read
; 1428           1790  5                    SET_IPL (0);                               ! Drop IPL before returning error status
; 1429           1791  5                    RETURN INS$_INTRNLERR;
; 1430           1792  4                    END;
; 1431           1793  4
; 1432           1794  4                [1] :        ! Greater than,
; 1433           1795  5                    BEGIN
; 1434           1796  5                    PRVKFE = .CMPKFE;
; 1435           1797  5                    CMPKFE = .CMPKFE [KFE$L_KFELINK];
; 1436           1798  4                    END;
; 1437           1799  4                TES;
; 1438           1800  3            END;                       ! WHILE traversing KFD's ordered KFE list
; 1439           1801  3
; 1440           1802  3        !
; 1441           1803  3        !   Have traversed whole list.  If PRVKFE has been set to 0, then
; 1442           1804  3        !   it was inserted, else it goes at the end.
; 1443           1805  3        !
; 1444           1806  3        IF .PRVKFE NEQ 0
; 1445           1807  3        THEN
; 1446           1808  3            PRVKFE [KFE$L_KFELINK] = .KFE;
; 1447           1809  2        END;             ! Insert KFE in ordered KFE list
; 1448           1810  2
; 1449           1811  2    SET_IPL (0);
; 1450           1812  2
; 1451           1813  2    INS$GL_KFEADR = .KFE;                     ! Return new KFE address in case of /LOG
; 1452           1814  2
; 1453           1815  2    INS$CNVRT_KF_LOCK (LCK$K_PRMODE);         ! Convert exclusive to protected read
; 1454           1816  2                                              !  to allow image activations
; 1455           1817  2
; 1456           1818  2    RETURN TRUE;
; 1457           1819  1    END;                     ! Routine Enter_kfe



                                        0FFC 00000 ENTER_KFE:
                                                   .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 1609
                   5B 00000000G  00  9E 00002       MOVAB   SGN_B_KFHSHSIZ, R11                            ;
                   5A      FE98  CF  9E 00009       MOVAB   ALLOC_PAGED, R10                               ;
                   59 00000000G  00  9E 0000E       MOVAB   INS$CNVRT_KF_LOCK, R9                          ;
                   58 00000000G  00  9E 00015       MOVAB   KFPB, R8                                       ;
                   5E                08  C2 0001C       SUBL2   #8, SP                                         ;
                                     05  DD 0001F       PUSHL   #5                                             ; 1642
                   69                01  FB 00021       CALLS   #1, INS$CNVRT_KF_LOCK                          ;
                   12                02  DA 00024       MTPR    #2, #18                                        ; 1645
```

```
                                                              I 16
INSCREATE                                                16-Sep-1984 01:49:49    VAX-11 Bliss-32 V4.0-742                 Page  43
V04-000         Enter_kfe  Enter the KFE into the hash table an 14-Sep-1984 12:35:36    [INSTAL.SRC]INSCREATE.B32;1             (14)

                                          5E  DD 00027        PUSHL   SP                                      ; 1646
                          52      04      AC  D0 00029        MOVL    KFE_TMP, R2                             ;
                          7E      08      A2  3C 0002D        MOVZWL  8(R2), -(SP)                            ;
                          6A              02  FB 00031        CALLS   #2, ALLOC_PAGED                         ;
                          39              50  E9 00034        BLBC    STATUS, 1$                              ;
                          57              6E  D0 00037        MOVL    KFE, R7                                 ; 1647
            67            62      08      A2  28 0003A        MOVC3   8(R2), (R2), (R7)                       ;
                                          68  D5 0003F        TSTL    KFPB                                    ; 1649
                                          44  12 00041        BNEQ    2$                                      ;
                                          58  DD 00043        PUSHL   R8                                      ; 1655
                                          10  DD 00045        PUSHL   #16                                     ;
                          6A              02  FB 00047        CALLS   #2, ALLOC_PAGED                         ;
                          23              50  E9 0004A        BLBC    STATUS, 1$                              ;
                          56              68  D0 0004D        MOVL    KFPB, R6                                ; 1656
   10       00            6E              00  2C 00050        MOVC5   #0, (SP), #0, #16, (R6)                 ;
                                          66     00055                                                        ;
                    08    A6              10  B0 00056        MOVW    #16, 8(R6)                              ; 1657
                    0A    A6      44      8F  90 0005A        MOVB    #68, 10(R6)                             ; 1658
                    10    AC              56  D0 0005F        MOVL    R6, NEWKFD_INSERT_ADR                   ; 1665
                                  04      A6  9F 00063        PUSHAB  4(R6)                                   ; 1670
                          50              6B  9A 00066        MOVZBL  SGN_B_KFHSHSIZ, R0                      ;
            7E            50              02  78 00069        ASHL    #2, R0, -(SP)                           ;
                          6A              02  FB 0006D        CALLS   #2, ALLOC_PAGED                         ;
                          6E              50  E9 00070 1$:    BLBC    STATUS, 7$                              ;
                          50              68  D0 00073        MOVL    KFPB, R0                                ; 1671
                          51              6B  9A 00076        MOVZBL  SGN_B_KFHSHSIZ, R1                      ;
                    0E    A0              51  B0 00079        MOVW    R1, 14(R0)                              ;
                          51              04  C4 0007D        MULL2   #4, R1                                  ; 1672
   51       00            6E              00  2C 00080        MOVC5   #0, (SP), #0, R1, @4(R0)                ;
                                  04      B0     00085                                                        ;
                          54              68  D0 00087 2$:    MOVL    KFPB, R4                                ; 1675
                          51      04      A4  D0 0008A        MOVL    4(R4), HSHTAB                           ;
                          50      08      AC  D0 0008E        MOVL    HSHIDX, R0                              ; 1685
                          56          6140    DE 00092        MOVAL   (HSHTAB)[R0], PRVKFE                    ;
                          55          6140    D0 00096        MOVL    (HSHTAB)[R0], CMPKFE                    ; 1686
                                          26  13 0009A 3$:    BEQL    5$                                      ; 1687
                          51      36      A7  9A 0009C        MOVZBL  54(R7), R1                              ; 1689
                          50      36      A5  9A 000A0        MOVZBL  54(CMPKFE), R0                          ; 1690
   50       20      37    A7              51  2D 000A4        CMPC5   R1, 55(R7), #32, R0, 55(CMPKFE)         ;
                                  37      A5     000AA                                                        ;
                                          0C  1A 000AC        BGTRU   4$                                      ;
                          67              66  D0 000AE        MOVL    (PRVKFE), (R7)                          ; 1704
                          66              57  D0 000B1        MOVL    R7, (PRVKFE)                            ; 1705
                                          56  D4 000B4        CLRL    PRVKFE                                  ; 1706
                                          55  D4 000B6        CLRL    CMPKFE                                  ; 1707
                                          E0  11 000B8        BRB     3$                                      ; 1689
                          56              55  D0 000BA 4$:    MOVL    CMPKFE, PRVKFE                          ; 1712
                          55              65  D0 000BD        MOVL    (CMPKFE), CMPKFE                        ; 1713
                                          D8  11 000C0        BRB     3$                                      ; 1687
                                          56  D5 000C2 5$:    TSTL    PRVKFE                                  ; 1722
                                          03  13 000C4        BEQL    6$                                      ;
                          66              57  D0 000C6        MOVL    R7, (PRVKFE)                            ; 1724
                                  0C      A4  B6 000C9 6$:    INCW    12(R4)                                  ; 1727
                    04    AE      0C      A7  D0 000CC        MOVL    12(R7), KFD                             ; 1729
                                          2D  12 000D1        BNEQ    9$                                      ; 1730
                                  04      AE  9F 000D3        PUSHAB  KFD                                     ; 1733
                          52      0C      AC  D0 000D6        MOVL    NEWKFD, R2                              ;
```

```
                              7E          08    A2  3C 000DA          MOVZWL   8(R2), -(SP)                                      :
                              6A                02  FB 000DE          CALLS    #2, ALLOC_PAGED                                   :
                              01                50  E8 000E1  7$:     BLBS     STATUS, 8$                                        :
                                                    04 000E4          RET                                                        :
                04    BE      62          08    A2  28 000E5  8$:     MOVC3    8(R2), (R2), @KFD                                 : 1734
                        0C    A7          04    AE  D0 000EB          MOVL     KFD, 12(R7)                                       : 1735
                        04    BE          10    BC  D0 000F0          MOVL     @NEWKFD_INSERT_ADR, @KFD                          : 1739
                        10    BC          04    AE  D0 000F5          MOVL     KFD, @NEWKFD_INSERT_ADR                           : 1740
                              50                68  D0 000FA          MOVL     KFPB, R0                                          : 1742
                                          0C    A0  B6 000FD          INCW     12(R0)                                            :
                              50          04    AE  D0 00100  9$:     MOVL     KFD, R0                                           : 1745
                                          0C    A0  B6 00104          INCW     12(R0)                                            :
                                          04    A0  D5 00107          TSTL     4(R0)                                             : 1750
                                                06  12 0010A          BNEQ     10$                                               :
                        04    A0                57  D0 0010C          MOVL     R7, 4(R0)                                         : 1755
                                                55  11 00110          BRB      15$                                               :
                              55                50  D0 00112  10$:    MOVL     R0, PRVKFE                                        : 1766
                              54          04    A0  D0 00115          MOVL     4(R0), CMPKFE                                     : 1770
                                                44  13 00119  11$:    BEQL     14$                                               : 1771
                              51          36    A7  9A 0011B          MOVZBL   54(R7), R1                                        : 1773
                              50          36    A4  9A 0011F          MOVZBL   54(CMPKFE), R0                                    : 1774
          50          20      37    A7          51  2D 00123          CMPC5    R1, 55(R7), #32, R0, 55(CMPKFE)                   :
                                          37    A4     00129                                                                     :
                                                29  1A 0012B          BGTRU    13$                                               :
                                                0E  1E 0012D          BGEQU    12$                                               :
                        04    A7                54  D0 0012F          MOVL     CMPKFE, 4(R7)                                     : 1780
                        04    A5                57  D0 00133          MOVL     R7, 4(PRVKFE)                                     : 1781
                                                55  D4 00137          CLRL     PRVKFE                                            : 1782
                                                54  D4 00139          CLRL     CMPKFE                                            : 1783
                                                DC  11 0013B          BRB      11$                                               : 1773
                  00000000G   00        0000'   CF  9E 0013D  12$:    MOVAB    DUPINKFD_ERR_DSC, INS$L_INTRNLERR                 : 1788
                                                03  DD 00146          PUSHL    #3                                                : 1789
                              69                01  FB 00148          CALLS    #1, INS$CNVRT_KF_LOCK                             :
                              12                00  DA 0014B          MTPR     #0, #18                                           : 1790
                              50  00000000G     8F  D0 0014E          MOVL     #INS$_INTRNLERR, R0                               : 1791
                                                04  00155             RET                                                        :
                              55                54  D0 00156  13$:    MOVL     CMPKFE, PRVKFE                                    : 1796
                              54          04    A4  D0 00159          MOVL     4(CMPKFE), CMPKFE                                 : 1797
                                                BA  11 0015D          BRB      11$                                               : 1771
                                                55  D5 0015F  14$:    TSTL     PRVKFE                                            : 1806
                                                04  13 00161          BEQL     15$                                               :
                        04    A5                57  D0 00163          MOVL     R7, 4(PRVKFE)                                     : 1808
                              12                00  DA 00167  15$:    MTPR     #0, #18                                           : 1811
                  00000000G   00                57  D0 0016A          MOVL     R7, INS$GL_KFEADR                                 : 1813
                                                03  DD 00171          PUSHL    #3                                                : 1815
                              69                01  FB 00173          CALLS    #1, INS$CNVRT_KF_LOCK                             :
                              50                01  D0 00176          MOVL     #1, R0                                            : 1818
                                                04  00179             RET                                                        : 1819

; Routine Size:  378 bytes,    Routine Base:  $CODE$ + 081E


; 1458          1820  1
; 1460          1821  1 %SBTTL 'Verify_channel  Is the file on the system device';
; 1461          1822  1
; 1462          1823  1 ROUTINE  VERIFY_CHANNEL (CHAN, RET_CCB_ADR) =
; 1463          1824  2 BEGIN
; 1464          1825  2 !+++
; 1465          1826  2 !
; 1466          1827  2 !  FUNCTIONAL DESCRIPTION:
; 1467          1828  2 !
; 1468          1829  2 !       Given the channel number, return the address of the
; 1469          1830  2 !       Channel Control Block.
; 1470          1831  2 !
; 1471          1832  2 !       CHAN            Channel number
; 1472          1833  2 !       RET_CCB_ADR     Longword in which to return CCB address
; 1473          1834  2 !---
; 1474          1835  2 LOCAL
; 1475          1836  2     STATUS;
; 1476          1837  2 GLOBAL REGISTER
; 1477          1838  2     CCB = 1;
; 1478          1839  2 MAP
; 1479          1840  2     CCB : REF BBLOCK;
; 1480          1841  2 BIND
; 1481          1842  2     RET_CCB = .RET_CCB_ADR;
; 1482          1843  2
; 1483          1844  2 !
; 1484          1845  2 !   Obtain the Channel Control Block
; 1485          1846  2 !
; 1486          1847  2 STATUS = IOC$VERIFYCHAN (.CHAN);
; 1487          1848  2 RET_CCB = .CCB;
; 1488          1849  2 RETURN .STATUS;
; 1489          1850  1 END;                    ! Routine Verify_channel



                                                        0FFC 00000 VERIFY_CHANNEL:
                                                                      .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11             ; 1823
                                  50        04   AC  D0 00002         MOVL    CHAN, R0                                         ; 1847
                                     00000000G   00  16 00006         JSB     IOC$VERIFYCHAN                                   ;
                              08   BC            51  D0 0000C         MOVL    CCB, @RET_CCB_ADR                                ; 1848
                                                     04 00010         RET                                                      ; 1850

; Routine Size:  17 bytes,    Routine Base:  $CODE$ + 0998


; 1490          1851  1
; 1492          1852  1 %SBTTL 'Check_shmident  Is the section in shared memory';
; 1493          1853  1
; 1494          1854  1 ROUTINE  CHECK_SHMIDENT (GBLNAMDSC, RET_IN_SHRMEM) =
; 1495          1855  2 BEGIN
; 1496          1856  2 !+++
; 1497          1857  2 !
; 1498          1858  2 !   FUNCTIONAL DESCRIPTION:
; 1499          1859  2 !
; 1500          1860  2 !        Check to see if the global section name translates to a name
; 1501          1861  2 !        which would place it in shared memory.
; 1502          1862  2 !
; 1503          1863  2 !---
; 1504          1864  2 LOCAL
; 1505          1865  2     NAM_DSC : BBLOCK [DSC$C_S_BLN],
; 1506          1866  2     SHRMEMNAM_DSC : BBLOCK [DSC$C_S_BLN],
; 1507          1867  2     SHRMEMNAM_BUF : BBLOCK [15],
; 1508          1868  2     GSDNAM_DSC : BBLOCK [DSC$C_S_BLN],
; 1509          1869  2     GSDNAM_BUF : BBLOCK [43],
; 1510          1870  2     STATUS;
; 1511          1871  2
; 1512          1872  2 GLOBAL REGISTER
; 1513          1873  2     SHRMEMNAM = 10,
; 1514          1874  2     GSDNAM = 11;
; 1515          1875  2 BIND
; 1516          1876  2     IN_SHARED_MEM = RET_IN_SHRMEM;
; 1517          1877  2
; 1518          1878  2 CH$MOVE (DSC$C_S_BLN, .GBLNAMDSC, NAM_DSC);            ! Copy the descriptor
; 1519          1879  2 NAM_DSC [DSC$W_LENGTH] = .NAM_DSC [DSC$W_LENGTH] - 4;   ! Drop the _000
; 1520          1880  2 SHRMEMNAM_DSC = 0;                                      ! Zero length
; 1521          1881  2 SHRMEMNAM_DSC [DSC$W_LENGTH] = 15;                      ! Zero length
; 1522          1882  2 SHRMEMNAM_DSC [DSC$A_POINTER] = SHRMEMNAM_BUF;          ! Set pointer to buffer on stack
; 1523          1883  2 SHRMEMNAM = SHRMEMNAM_DSC;                              ! Place address of descriptor in R10
; 1524          1884  2 GSDNAM_DSC = 0;                                         ! Zero the length
; 1525          1885  2 GSDNAM_DSC [DSC$W_LENGTH] = 43;                         ! Zero the length
; 1526          1886  2 GSDNAM_DSC [DSC$A_POINTER] = GSDNAM_BUF;                ! Set pointer to buffer on stack
; 1527          1887  2 GSDNAM = GSDNAM_DSC;                                    ! Place address of descriptor in R11
; 1528          1888  2
; 1529          1889  2 STATUS = MMG$GSDTRNLOG ( NAM_DSC );                     ! Translate logical name to see if section name has
; 1530          1890  3 .IN_SHARED_MEM = (IF .SHRMEMNAM_DSC [DSC$W_LENGTH] NEQ 0
; 1531          1891  3                 THEN TRUE                               ! Return true if there was a shared memory name tran
; 1532          1892  2                 ELSE FALSE);
; 1533          1893  2 RETURN .STATUS;
; 1534          1894  1 END;                        ! Routine Check_shmident



                                                OFFC 00000 CHECK_SHMIDENT:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11              ; 1854
                        5E      AC   AE  9E 00002       MOVAB   -84(SP), SP                                            ;
      4C    AE     04   BC           08  28 00006       MOVC3   #8, @GBLNAMDSC, NAM_DSC                                ; 1878
                   4C   AE           04  A2 0000C       SUBW2   #4, NAM_DSC                                            ; 1879
                                44   AE  D4 00010       CLRL    SHRMEMNAM_DSC                                          ; 1880
                   44   AE           0F  B0 00013       MOVW    #15, SHRMEMNAM_DSC                                     ; 1881
                   48   AE      34   AE  9E 00017       MOVAB   SHRMEMNAM_BUF, SHRMEMNAM_DSC+4                         ; 1882
                        5A      44   AE  9E 0001C       MOVAB   SHRMEMNAM_DSC, SHRMEMNAM                               ; 1883
```

```
                                  2C   AE  D4 00020           CLRL    GSDNAM_DSC                                        ; 1884
                     2C   AE           2B  B0 00023           MOVW    #43, GSDNAM_DSC                                   ; 1885
                     30   AE           6E  9E 00027           MOVAB   GSDNAM_BUF, GSDNAM_DSC+4                          ; 1886
                          5B      2C   AE  9E 0002B           MOVAB   GSDNAM_DSC, GSDNAM                                ; 1887
                          59      4C   AE  9E 0002F           MOVAB   NAM_DSC, R9                                       ; 1889
                             00000000G 00  16 00033           JSB     MMG$GSDTRNLOG                                     ;
                                  44   AE  B5 00039           TSTW    SHRMEMNAM_DSC                                     ; 1890
                                       05  13 0003C           BEQL    1$                                                ;
                          51           01  D0 0003E           MOVL    #1, R1                                            ;
                                       02  11 00041           BRB     2$                                                ;
                                       51  D4 00043 1$:       CLRL    R1                                                ;
                     08   BC           51  D0 00045 2$:       MOVL    R1, @IN_SHARED_MEM                                ;
                                       04     00049           RET                                                       ; 1894

; Routine Size:  74 bytes,    Routine Base:  $CODE$ + 09A9


; 1535          1895  1
```

```
1 %SBTTL 'INS$BLD_GBLSECNAM  Build the global section name string';
1
1 GLOBAL ROUTINE  INS$BLD_GBLSECNAM (GBLNAMDSC) =
2 BEGIN
2 !+++
2 !
2 !  FUNCTIONAL DESCRIPTION:
2 !
2 !       Build the global section name.  If the name does not exist,
2 !       get the root from the NAM block and append _001.  If it does
2 !       exist, increment the suffix.
2 !
2 !---
2 LOCAL
2     NAMSTR : REF BBLOCK,
2     PTR;
2 BIND
2     GBLNAM_SUFFIX = UPLIT (%ASCII '_001') : VECTOR [,BYTE];     ! First suffix
2 MAP
2     GBLNAMDSC : REF BBLOCK;
2
2 NAMSTR = .GBLNAMDSC [DSC$A_POINTER];                             ! Pointer to last global section name, or ze
2 IF .GBLNAMDSC [DSC$W_LENGTH] EQL 0                               ! If the name is zeroed then this is the fir
2 THEN
3     BEGIN
3     GBLNAMDSC [DSC$W_LENGTH] = .INS$G_KFENAM [NAM$B_NAME] + 4;  ! Size is filename length plus 4 for _001
3     PTR = .NAMSTR;                                              ! Point past count byte
3     PTR = CH$MOVE (.INS$G_KFENAM [NAM$B_NAME], .INS$G_KFENAM [NAM$L_NAME], .PTR);      ! Move filename in
3     CH$MOVE (4, GBLNAM_SUFFIX, .PTR);                           ! Move _001 suffix in
3     END
2 ELSE
3     BEGIN                                                       ! Name has already been built, increment the
3     PTR = .NAMSTR + .GBLNAMDSC [DSC$W_LENGTH] - 1;              ! Locate last digit of suffix number
3     WHILE ( .(.PTR) <0,8> NEQ %C'_' ) DO                        ! Don't want carry to clobber the '_' separa
4         BEGIN
4         (.PTR) <0,8> = .(.PTR) <0,8> + 1;                       ! Add one to suffix number
4
5         IF ( .(.PTR) <0,8> GTR %C'9' )                          ! If that raises it over '9' than make it a
4         THEN
5             BEGIN
5             (.PTR) <0,8> = %C'0';                               ! Make '9' into a '0'
5             PTR = .PTR - 1;                                     ! Move to next highest decimal place
5             END
4         ELSE
4             RETURN TRUE;
3         END;
2     END;
2
2 RETURN TRUE;
1 END;                          ! Routine INS$BLD_GBLSECNAM


                                                  .PSECT  $PLIT$,NOWRT,NOEXE,2

                     31  30  30  5F   0003C P.AAE:   .ASCII  \_001\                          ;
```

```
                                                    GBLNAM_SUFFIX=      P.AAE

                                                          .PSECT  $CODE$,NOWRT,2

                                         003C 00000       .ENTRY  INS$BLD_GBLSECNAM, Save R2,R3,R4,R5          ; 1898
                        52        04   AC  D0 00002       MOVL    GBLNAMDSC, R2                                ; 1917
                        50        04   A2  D0 00006       MOVL    4(R2), NAMSTR                                ;
                        51             62  3C 0000A       MOVZWL  (R2), R1                                     ; 1918
                                       20  12 0000D       BNEQ    1$                                           ;
                        51  00000000G  00  9A 0000F       MOVZBL  INS$G_KFENAM+59, R1                          ; 1921
              62        51             04  A1 00016       ADDW3   #4, R1, (R2)                                 ;
                        53             50  D0 0001A       MOVL    NAMSTR, PTR                                  ; 1922
                        50  00000000G  00  D0 0001D       MOVL    INS$G_KFENAM+76, R0                          ; 1923
              63        60             51  28 00024       MOVC3   R1, (R0), (PTR)                              ;
                        63      0000'  CF  D0 00028       MOVL    GBLNAM_SUFFIX, (PTR)                         ; 1924
                                       19  11 0002D       BRB     3$                                           ; 1918
                        53     FF A140  9E 0002F 1$:      MOVAB   -1(R1)[NAMSTR], PTR                          ; 1928
                   5F   8F             63  91 00034 2$:   CMPB    (PTR), #95                                   ; 1929
                                       0E  13 00038       BEQL    3$                                           ;
                                       63  96 0003A       INCB    (PTR)                                        ; 1931
                        39             63  91 0003C       CMPB    (PTR), #57                                   ; 1933
                                       07  1B 0003F       BLEQU   3$                                           ;
                        63             30  90 00041       MOVB    #48, (PTR)                                   ; 1936
                                       53  D7 00044       DECL    PTR                                          ; 1937
                                       EC  11 00046       BRB     2$                                           ; 1933
                        50             01  D0 00048 3$:   MOVL    #1, R0                                       ; 1944
                                       04     0004B       RET                                                  ; 1945

; Routine Size:  76 bytes,    Routine Base:  $CODE$ + 09F3

; 1587         1946  1
; 1589          1947  1 END               ! Module inscreate
; 1590          1948  0 ELUDOM



                                                                                      .EXTRN  LIB$SIGNAL

;                                         PSECT SUMMARY
;
;       Name                         Bytes                          Attributes
;
;   $OWN$                                 16  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $PLIT$                                64  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                              2623  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                                 Library Statistics
;
;                                                  -------- Symbols --------        Pages         Processing
;       File                                       Total    Loaded  Percent         Mapped        Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1                  18619      129        0          1000            00:01.9




;                                          COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:INSCREATE/OBJ=OBJ$:INSCREATE MSRC$:INSCREATE/UPDATE=(ENH$:INSCREATE)

; Size:           2623 code + 80 data bytes
; Run Time:          00:51.5
; Elapsed Time:      02:45.1
; Lines/CPU Min:       2268
; Lexemes/CPU-Min: 19859
; Memory Used:  488 pages
; Compilation Complete
```

0188 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

INPSMB
MAP

INSDEF
SDL

INPSMBMSG
LIS

RSXLBLDF
SDL

INSCREATE
LIS

INITIO
LIS

INSTAL

INSTALL
MAP

INSCMD
CLD

INPSMBCLD
CLD

INSPREFIX
REQ

INPSMB
LIS

INSOLDCMD
CLD

INIVOL
LIS

RDHOME
LIS

INPSMB

INPSMBCLD
LIS

INSCMD
LIS

0189 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

INSMAIN
LIS

INSDELETE
LIS

INSOLDCMD
LIS

INSPRSOLD
LIS

INSGLOBAL
LIS

INSKFSCAN
LIS

INSLIST
LIS

INSMSG
LIS